



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1984-09

Development of graphical Pole-zero,
Root-Locus, Bode, Nyquist, and Nichols
responses using the OPTSYSX Program.

Laptas, Michael Henry

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/19260>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



<http://www.nps.edu/library>

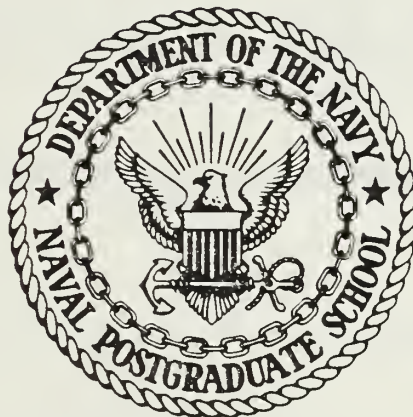
Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

DEVELOPMENT OF GRAPHICAL POLE-ZERO,
ROOT-LOCUS, BODE, NYQUIST, AND NICHOLS
RESPONSES USING THE OPTSYSX PROGRAM

by

Michael Henry Laptas

September 1984

Thesis Advisor:

Daniel J. Collins

Approved for public release; distribution unlimited

T222961

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development of Graphical Pole-Zero, Root-Locus, Bode, Nyquist, and Nichols Responses Using the OPTSYSX Program		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1984
7. AUTHOR(s) Michael Henry Laptas		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1984
		13. NUMBER OF PAGES 219
		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
OPTSYSX Pole-Zero Root-Locus Bode Nyquist Nichols Transfer function Open Loop Noise Compensator		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>This thesis discusses the modification of and additions to an existing Optimal Systems Control FORTRAN Program (OPTSYS) originally obtained from Professor Arthur E. Bryson of Stanford University. This program has been subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden, and additions by Commander Harry A. Diel provide the user with a highly accurate graphic time response to a system designed using the OPTSYSX program.</p>		

The addition of the FORTRAN program OPTGRAPH gives the user the capability to obtain classical analysis (Pole-Zero Map, Root Locus, Bode, Nyquist, and Nichols) of transfer functions, calculated for a system designed with the OPTSYSX program. The OPTGRAPH program uses high resolution precision plotting software to provide the user with highly accurate frequency response plots.

This series of programs permits users to rapidly carry out simulation, analysis, and design of Optimal Systems Control problems in a totally interactive mode.

Approved for public release; distribution unlimited.

Development of Graphical Pole-Zero, Root-Locus, Bode,
Nyquist, and Nichols Responses using the
OPTSYSX Program

by

Michael H. Laptas
Lieutenant, United States Navy
B.S., Purdue University, Lafayette, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1984

ABSTRACT

This thesis discusses the modification of and additions to an existing Optimal Systems Control FORTRAN Program (OPTSYS) originally obtained from Professor Arthur E. Bryson of Stanford University. This program has been subsequently redesigned to run interactively on the IBM 3033 VM/CMS by Lieutenant Commander John G. Hoden, and additions by Commander Harry A. Diel provide the user with a highly accurate graphic time response to a system designed using the OPTSYSX program.

The addition of the FORTRAN program OPTGRAPH gives the user the capability to obtain classical analysis (Pole-Zero Map, Root Locus, Bode, Nyquist, and Nichols) of transfer functions, calculated for a system designed with the OPTSYSX program. The OPTGRAPH program uses high resolution precision plotting software to provide the user with highly accurate frequency response plots.

This series of programs permits users to rapidly carry out simulation, analysis, and design of Optimal Systems Control problems in a totally interactive mode.

TABLE OF CONTENTS

I.	INTRODUCTION	9
II.	THE OPTSYSX COMPUTER PROGRAM	11
	A. BACKGROUND	11
	B. OVERVIEW	11
	C. OPTSYSX MCDIFICATIONS	12
	D. OPTSYS EXEC MODIFICATIONS	13
III.	THE OPGRAPH PROGRAM	15
	A. PROGRAM OVERVIEW	15
	1. Program Language	15
	2. Graphics Package	15
	3. Library Subroutines	15
	4. Program Composition	16
	B. GENERAL PROGRAM OPERATION	16
	1. User Protection Features	16
	2. Large Order System	17
	C. SYSTEM/MODEL DESCRIPTION	18
	D. PROGRAM ORGANIZATION	19
	1. Main Program	19
	2. Program Sequencing	20
	3. Graphic and/or Tabular System Response	21
	E. INTERACTIVE EXAMPLES	23
	F. CHARACTERISTIC EQUATION ANALYSIS	24
	1. Pole-Zero Map	24
	2. Pole-Zero Map (Interactive Example)	25
	3. Root-Locus	33
	4. Root-Locus (Interactive Example)	34

G.	FREQUENCY RESPONSE ANALYSIS	43
1.	Bode Response	44
2.	Bode Response (Interactive Example) . . .	45
3.	Nyquist Response	51
4.	Nyquist Response (Interactive Example) . .	52
5.	Nichols Response	52
6.	Nichols Response (Interactive Example) . .	52
IV.	CONCLUSIONS AND RECOMMENDATIONS	59
A.	CONCLUSIONS	59
B.	RECOMMENDATIONS	60
1.	Root Finding Routines	60
2.	Program Memory Requirements	61
3.	Further Modifications	61
	APPENDIX A: OPTSYS EXEC LISTING	62
	APPENDIX B: OPTSYSX PROGRAM LISTING	71
	APPENDIX C: OPTGRAPH PROGRAM LISTING	160
	APPENDIX D: OPGRAPH LISTING	215
	LIST OF REFERENCES	217
	BIBLIOGRAPHY	218
	INITIAL DISTRIBUTION LIST	219

LIST OF FIGURES

3.1	Pole-Zero Map Example	53
3.2	Root Locus Response Example	54
3.3	Bode Response Example (Magnitude)	55
3.4	Bode Response Example (Phase)	56
3.5	Nyquist Response Example	57
3.6	Nichols Response Example	58

SYMBOLS

A = State (Ns,Ns) or Output (No,No) Weighting Matrix
B = Control (Nc,Nc) Weighting Matrix
C = Control Gain Matrix (Nc,Ns)
D = Control (No,Nc) or Noise (No,Ng) Feedforward Matrix
F = Open-Loop Dynamics Matrix (Ns,Ns)
G = Control Distribution Matrix (Ns,Nc)
GAM = State Disturbance Distribution Matrix (Ns,Ng)
H = Measurement Scaling Matrix (No,Ns)
K = Estimator Gain Matrix (Ns,Nob)
Nc = Number of Controls
Ng = Number of Process Noise Sources
Ns = Number of States
No = Number of Observations or Measurements
Q = White Process Noise Covariance Matrix (Ng,Ng)
R = White Meas. Noise Covariance Matrix (No,No)
S = Steady-State Covariance Matrix of Control (Nc,Nc)
u = Control Vector (Nc,1)
v = White Measurement Noise Vector (No,1), with Zero Mean and Covariance Matrix R
w = White Process Noise Vector (Ng,1), with Zero Mean and Covariance Matrix Q
w0 = Constant Disturbance Vector (Ng,1)
x = State Vector (Ns,1)
xdot = Derivative of State Vector (Ns,1)
xe = Estimate of State Vector (Ns,1)
xedot = Derivative of Estimate of State Vector (Ns,1)
y = Output Vector (No,1)
z = Measurement Vector (No,1)

I. INTRODUCTION

The purpose of this thesis is to describe the modification and additions to the existing FORTRAN program (OPTSYS) which is used in the study and application of Optimal Systems Control theory.

The Optimal Systems control program was originally developed by Hall [Ref. 1] to support his research in rotary-wing aircraft control systems. Later program modifications were made by Walker [Ref. 2] and Liu [Ref. 3] of Stanford University, and are designated OPTSYS 4 and OPTSYS 5 respectively. OPTSYS modifications made by Hoden [Ref. 4] were primarily devoted to creating a user-friendly interactive version (OPTSYSX) of the OPTSYS 4 program. The latest modifications by Diel [Ref. 5] allowed the user to save matrices for subsequent runs of the OPTSYSX program, and formed data file sets for the time response program OPTCALC.

The intent of this thesis work was to develop an interactive program to plot the Pole-Zero map, Root-Locus, Bode, Nyquist, and Nichols responses to the open loop, closed loop noise, and compensator transfer functions of a State Variable Control System which has been developed using OPTSYS Program. Minor modifications to the OPTSYSX Program were necessary to construct a data file sets for plotting the Pole-Zero Map, and in calculations for the Root-Locus, Bode, Nyquist, and Nichols plots.

It is assumed that the user is familiar with the basic concepts of Control Theory and Optimal Systems Design. The symbol/naming conventions of Bryson [Ref. 6]. are used in the discussion of program operations and descriptions of problems using the OPTSYS System.

An overview of the OPTSYSX Program capabilities and of modifications to the existing programs is presented first. This is followed by a description of the program (OPTGRAPH), which was developed to plot the Pole-Zero Map and to perform calculations for the Root-Locus, Bode, Nyquist, and Nichols plots.

This work concludes with examples of various types of problems demonstrated in the interactive mode, including a copy of each terminal session with the final results. Complete program listings for the OPTSYS EXEC program, OPTSYSX program, and OPTGRAPH program are included in appendices A, B, and C respectively.

II. THE OPTSYSX COMPUTER PROGRAM

A. BACKGROUND

OPTSYSX is a double-precision, interactive FORTRAN program employing modern control theory analysis techniques. Its capabilities include the calculation of the open-loop eigensystem, and the stationary closed loop system; the synthesis of regulators and filters; along with power spectral density, and modal distribution computations. The modifications introduced to the OPTSYSX program by this thesis work do not affect the program's original capabilities.

B. OVERVIEW

OPTSYSX is an extremely large and complex program with over 3000 lines of code. To use the program in its small version (dimensioned for a 32 X 32 "F", "G", and "H" matrices), the user must extend his Virtual Machine (VM) memory capacity beyond the default VM memory capacity of 720 kilobytes to 1024 kilobytes.

Any significant increase in the OPTSYSX program size, with the the resulting requirement for additional memory capacity, would cause user difficulties. For this reason the task of obtaining the Pole-Zero maps, Root-Locus, and Bode, Nyquist, Nichols plots for the open loop, noise and compensator transfer functions was relegated to a separate program (OPTGRAPH). Three data files (OPTGROL DATA, OPTGRNO DATA, and OPTGRCM DATA), containing the open loop, noise, and compensator transfer functions, are formed by the OPTSYS program to be transferred to the OPTGRAPH program. The OPTSYS EXEC program provides an interface between the

OPTSYSX program, the time response programs (OPTCALC and OPTPICT), and the OPTGRAPH program.

C. OPTSYSX MODIFICATIONS

OPTSYSX program modifications consist of the addition of write statements, three flags, and a short routine to interpret user inputs.

Write statements, to input data to the OPTGROL, CPTGRNO, and CPTGRCM DATA files were added to the main program, subroutine INNER, subroutine ZEROS, and subroutine POLES. System information consisting of number of states (Ns), number of controls (Nc), number of measurements (No), number of process noise sources (Ng), type transfer function (ITFX), Markov Parameter (IE), and two flags to be used by the CPTSYS EXEC and the OPTGRAPH programs is inputted from the main program. The write statements added to the subroutine INNER, and subroutine POLES input the poles for the open loop and noise transfer functions, and compensator transfer function respectively. The zeros, the numerator order, the gain, and the input and output numbers for all three transfer functions are obtained from the subroutine ZEROS.

The first of the two flags, input with the system information from the main program to the three data files, serves as a marker for the CPTSYS EXEC to locate the correct data line for reading system inputs. The second flag, which is set by the routine to interpret user inputs, serves as a signal for the OPTSYS EXEC to either load the OPTGRAPH program if any one of the three transfer functions was calculated by the OPTSYSX program, and abort loading the OPTGRAPH program if none of the three transfer functions were calculated. The third flag (ITFX) is an existing OPTSYSX program flag. The flag (ITFX) is passed from the

subroutine TF to the subroutine ZEROS to identify the calculation of either an open loop, noise, or compensator transfer function zeros.

The short routine added to the main program of OPTSYSX reads the flags ITF1, ITF2, ITF3, and IRET. The flag sent to the OPTGROL, OPTGRNO, or OPTGRM DATA files is set to allow loading of transfer function data file to the OPTGRAPH program if the user selects either of the transfer function options number two (calculate poles, residues, and zeros - ITF1 = 1, ITF2 = 1, or ITF3 = 1) number three (calculate poles, and zeros - ITF1 = 2, ITF2 = 2, or, ITF3 = 2) when calculating the open loop, noise, or compensator transfer functions. The selection of any other transfer function option will not calculate the transfer function zeros. The flag IRET is read to determine if the user desires to rerun a problem without exiting the OPTSYSX program. Should the user elect to rerun a problem without exiting OPTSYSX (IRET = 1), the three data files are set back to the beginning (REWIND) to accept the new problem system data.

D. OPTSYS EXEC MODIFICATIONS

The OPTSYS EXEC was written by Diel [Ref. 5] to interface between the OPTSYSX program and the time response programs OPTCALC and CPTPLOT. The modifications to the exec program were to add the option to select running the OPTGRAPH program to the exec menus and a short section to interface the OPTGRAPH program with the OPTSYSX program.

The added section to interface between the OPTSYSX and OPTGRAPH programs defined loader size (LDRTBLE), file definitions (FILEDEF), and text libraries (TXTLIB) for the OPTGRAPH program. The first line of the three data files is read to ensure that there has been at least one of the three transfer functions calculated by the OPTSYSX program prior to loading OPTGRAPH program.

The OPTSYS EXEC routine is also used to send plots to the VERSATEC plotter from the OPTPLOT program and the OPTGRAPH program.

III. THE OPGRAPH PROGRAM

A. PROGRAM OVERVIEW

OPTGRAPH is an interactive FORTRAN program which uses the transfer function poles and zeros calculated by OPTSYSX to calculate, tabulate, and plot the Pole-Zero Map, Bode, Nyquist, Nichols, and Root-Locus responses for either the open loop, closed loop noise, or compensator transfer functions.

1. Program Language

CPTGRAPH is programmed in FORTRAN following the conventions of the IBM System /360/370/ FORTRAN IV language. CPTGRAPH has been compiled and run under both FORTRAN IV (G1) and FORTRAN H (extended) compilers on the IBM 3033.

2. Graphics Package

This program uses the Display Integrated Software System and Plotting System Software (DISSPLA) developed and distributed by the Integrated Software Systems Corporation (ISSCC) of San Diego, California. The DISSPLA package is a library of FORTRAN subroutines using FORTRAN IV conventions.

3. Library Subroutines

CPTGRAPH uses the International Mathematical and Statistical Library (IMSL) subroutine ZRPOLY to calculate the roots of a polynomial equation. ZRPOLY is a double precision FORTRAN subroutine to which is input a polynomial with real coefficients (double precision) in terms of decreasing powers of the polynomial. The subroutine finds the roots of the polynomial and returns a double precision

complex array containing the roots. ZRPOLY is capable of accepting polynomials greater than 0 order and less than 101 order.

4. Program Composition

CPTGRAPH has one main program and 24 subroutines. The main program and its subroutines may be divided into four basic categories:

- 1) File Data Input
- 2) Interactive Data Input
- 3) Plot Setup and Sequencing
- 4) Calculation

A brief and general description of the program and its subroutines will follow in subsequent sections.

B. GENERAL PROGRAM OPERATION

OFTGRAPH was written to satisfy two specific objectives. The first is to create a program to be used as a instructional tool for students taking controls related courses, and the second is as method to assist in the analysis of actual large order control system problems.

To accomplish these objectives the emphasis while writing this program was to make the program as user oriented as possible by attempting to eliminate ambiguities and providing features which would protect the user from inadvertent wrong entries. An effort was also made to minimize the Virtual Machine (VM) memory required to execute the program to facilitate the analysis of a large order system.

1. User Protection Features

Three basic methods were used to protect the user from an abnormal program termination in the case of an inadvertent wrong entry. The first was to construct the

subroutines to display a summary of the user inputs and allowing changes prior to leaving the subroutine. For the case where the user input is expected to fall within a given range, the input is examined and the user is either allowed to continue for an input within the range or issued an error or warning and reasked the question for an input outside the range. The final method is used by the three subroutines which read the user input from the screen. The subroutines RDINT, RDREAL, and RDCHAR expect an integer input, real number input, and a logical "yes" or "no" respectively. In the event that a "null" line is inadvertently entered, the user is issued a warning and allowed another opportunity to enter the correct input before abnormally terminating the program. The entry of two "null" lines also offers the user the option to exit the OPTGRAPH program at other than normal program exit points. The subroutine RDCHAR also examines the input and issues a warning if the input is not either "yes" or "no".

2. Large Order System

The large order control problem being considered for analysis by the OPTSYSX and the OPTGRAPH programs is the X-29A aircraft longitudinal axis backup mode system. This system has a (98 X 98) "F" matrix, a (2 X 98) "H" matrix, and a (98 X 1) "G" matrix.

To accommodate this large order system the OPTGRAPH program was dimensioned to accept a system with a maximum of 99 states (Ns), 12 controls (Nc), 12 process noise sources (Ng), and 12 measurements (No). The Virtual Machine (VM) memory requirements for the program were minimized by the reuse of memory locations allocated for storage of calculated data after that data has been either tabulated or plotted. This method of memory reuse has allowed the OPTGRAPH program to be dimensioned to accommodate a large

order system and still operate with less than the 1024 KILOBYTES VM memory required to run OPTSYSX in its small version.

C. SYSTEM/MODEL DESCRIPTION

The system equations used by the OPTSYSX program for the calculation of the transfer functions are of the state variable form. The system equations are:

system model

$$\dot{x} = [F]*x + [G]*u + [GAM]*w \quad (3.1)$$

measurement equation

$$z = [H]*x + [D]*u + v \quad (3.2)$$

estimator equation

$$\dot{x}_e = [F]*x_e + [G]*u + [K]*(z - [H]*x_e) \quad (3.3)$$

open loop transfer function

$$[H]*[s[I] - [F]]^{-1}*[G] \quad (3.4)$$

closed loop noise transfer function

$$[H]*[s[I] - [F]]^{-1}*[GAM] \quad (3.5)$$

compensator transfer function from measurement to input

$$[C]*([s[I] - [F]] + [G]*[C] + [K]*[H])^{-1}*[K] \quad (3.6)$$

where

u = Control vector (Ns X 1)

w = White process noise vector ($N_g \times 1$)
 x = State vector ($N_s \times 1$)
 \dot{x} = Derivative of the state vector ($N_s \times 1$)
 y = Output vector ($N_o \times 1$)
 z = measurement vector ($N_o \times 1$)

 $[C]$ = control gain matrix (N_c, N_s)
 $[D]$ = control feed-forward distribution matrix (N_c, N_c)
of the control vector (u)
 $[F]$ = open-loop dynamics matrix (N_s, N_s)
(System matrix or Plant matrix)
 $[G]$ = control distribution matrix (N_s, N_c)
 $[GAM]$ = state disturbance distribution matrix (N_s, N_g)
 $[H]$ = measurement distribution matrix (N_o, N_s)
 $[I]$ = identity matrix (N_s, N_s)

D. PROGRAM ORGANIZATION

1. Main Program

The main program presents a menu allowing the user to select for analysis either the open loop transfer function, the closed loop noise transfer function, the compensator transfer function, or exit the OPTGRAPH program. The data file for the selected transfer function is examined by the main program to ensure that the file contains the required transfer function data. Should the data file be incomplete the program gives the user a warning message indicating the nature of the missing data and allows the user to select another option.

The main program menu:

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

Data is read from the selected file by the main program and formed into three arrays and seven integer variables. The exchange of data between the main program and the subroutines, and between the subroutines is by the extensive use of labeled common statements. The data transferred through the subroutine call statements is limited primarily to flags, and constants. The program moves to its primary sequencing subroutine (GRAPH) upon the completion of file data entry.

2. Program Sequencing

The subroutine GRAPH first presents the user with a summary of the the transfer system received from the data file. The user is then asked which transfer function he desires to to analyze using the conventions of the OPTSYSX program describing the transfer functions by input number and output number. A menu is presented which allows the user to select graphical and/or tabular system response (Pole-Zero map, Root-locus, Bode, Nyquist, or Nichols), select another transfer function, or exit the OPTGRAPH program. Upon completion of a graphical and/or tabular response option, or change of transfer function option, the program returns to the GRAPH menu to allow the user to choose another option or exit to the main program.

Subroutine GRAPH data summary and menu:

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS CONTAINS:

- | | |
|-----------------------|---|
| 1. NUMBER OF STATES = | 4 |
|-----------------------|---|

2. NUMBER OF CCNTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST, AND/
OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

OUTPUT # ?

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME TRANSFER
FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT OPTGRAPH

ENTER OPTICN NUMBER.

3. Graphic and/or Tabular System Response

The selection of a graphical and/or tabular system response option for an Pole-Zero map, Root-Locus, Bode, open loop Nyquist, or open loop Nichols calls the subroutines

PZERO, RTLO, BODE, NYQST, or NICHOL respectively. While the response of these subroutines is different, their organization is similar.

The subroutines PZERO, RTLO, BODE, NYQST, and NICHOL primarily act as sequencing and plot setup subroutines. The interactive questioning of the user for graphic/tabular response information and the system response calculation has been relegated to other subroutines. The subroutines PZERO, RTLO, BODE, NYQST, and NICHOL are divided into two major sections with the first being devoted to producing a graphical output and the second to producing a tabular output.

The user is presented a subroutine menu offering the option for graphic response, tabular response, or exiting the subroutine. The program moves to either the first section (graphic response), second section (tabular response) or back to the subroutine GRAPH menu depending on the option selection. The program returns to the subroutine menu upon completion of either the graphic response, or tabular response section, allowing the user to select another response option, or exit the subroutine.

Features that are common to the five subroutines will be described here and features that are unique to one of the subroutines will be described in the following section under the appropriate response heading.

For a graphic response, the user is given the choice of two printer options (TEK 618, or VERSATEC). With the selection of the TEK 618, the plot page size is defaulted to 11 inches by 8.5 inches, and the plot will be presented on the TEK 618 screen. The VERSATEC option gives the user the option to create up to a 21 inch by 21 inch plot (maximum for the VERSATEC printer) and causes a DISPLA METAFILE to be created. A scaling subroutine (PSCALE) scales the plot heading, legend, etc. in proportion to the page width selected. The VERSATEC option offers the the advantage of a

high quality print from the VERSATEC printer using the VERSATEC printer option in the OPTSYS EXEC. The creation of a DISSPLA METAFILE also gives the user (after he exits optsys) the opportunity to use printers other than the TEK 618 or VERSATEC. A major disadvantage with the VERSATEC option is that the user must exit the OPTGRAPH program before he can print a graphic system response. For this reason, it is recommended that the system response first be plotted on the TEK 618 prior to selecting the VERSATEC option.

For a tabular response, the user is presented a menu which gives him the choice of three devices (screen, printer, or disk) to send the tabular output. The user is asked if he desires to make any changes, after the tabular data has been calculated. If the answer is affirmative, he is presented a menu and allowed to make changes and rerun the tabular data routine.

Examples of these features are included in the following sections with the interactive examples for the system responses.

E. INTERACTIVE EXAMPLES

The open-loop, closed loop noise, and compensator transfer functions are handled identically by the OPTGRAPH program, with the only difference being the program and graph headings identifying the transfer function type. The frequency response programs for the Bode, Nyquist, and Nichols responses use identical setup and tabular data routines.

A good overall representation of the OPTGRAPH program capabilities will be demonstrated by single interactive terminal session examples for the pole-zero, root locus, and bode responses using selected transfer functions. Tabular

data examples will be provided for the Pole-Zero, Root Locus, and Bode responses, and graphic response examples will be provided the Pole-Zero, Root Locus, Bode, Nyquist, and Nichols responses.

F. CHARACTERISTIC EQUATION ANALYSIS

The subroutines PZERO and RTLO are similar in their basic formats. The pole and zero locations for the selected transfer function input and output numbers are read into two single dimension arrays. The extraneous zeros, calculated by the OPTSYSX program, are eliminated by comparing the zeros to the Markov parameter sent with the system information from the OPTSYSX program. For a graphical response, the user is asked to define the plot limits in terms of X-coordinates (real axis) and Y-coordinates (imaginary axis). Points for the plot which fall outside these limits will be ignored. System information (transfer function input number, output number, and (DC) gain) is listed in the graph legend.

1. Pole-Zero Map

The subroutine PZERO converts the double precision numbers for pole and zero locations to single precision for compatibility with the with the Graphics Package and plots the poles and zeros within the plot limits defined by the user for the graphical response routine. For tabular data response the user is given system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, and the pole and zero locations.

2. Pole-Zero Map (Interactive Example)

The following example of a simplified open loop transfer function for an airplane with an autopilot in the longitudinal mode was obtained from [Ref. 7,p.315].

The aircraft system is represented by:

$$[D] = [0]$$

$$[F] = \begin{bmatrix} 0. & 1. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 1. \\ 0. & 16. & -12. & -3. \end{bmatrix}$$

$$[G] = \begin{bmatrix} 0. \\ 0. \\ 1. \\ -2. \end{bmatrix}$$

$$[GAM] = [0]$$

$$[H] = [1. \quad 0. \quad 0. \quad 0.]$$

a. Example of Graphic Response (Terminal Session)

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 19:40:17

EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

1

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS
CONTAINS:

1. NUMBER OF STATES = 4
2. NUMBER OF CONTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,
AND? OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP

2. ROOT-LOCUS
3. EODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT
CPTGRAPH

ENTER OPTICN NUMBER.

1

POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

1

POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

PLOTTER SELECTION AND PAGE SIZE

(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618
2. VERSATEC

CHOCSE OPTION 1 OR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

6

WIDTH =

8

PLOTTING LIMITS FOR GRAPH

 X AXIS (REAL AXIS)

 X MINIMUM = ?

-15

 X MAXIMUM = ?

5

 Y AXIS (IMAGINARY AXIS)

 Y MINIMUM = ?

-5

 Y MAXIMUM = ?

5

PLOTTING LIMITS FOR GRAPH

 X AXIS (REAL AXIS)

 X MINIMUM = -15.00

 X MAXIMUM = 5.00

 Y AXIS (IMAGINARY AXIS)

 Y MINIMUM = -5.00

 Y MAXIMUM = 5.00

DO YOU DESIRE TO MAKE ANY CHANGES ?

 TYPE "YES" OR "NO".

n

PLOT HEADING

 HOW MANY LINES OF HEADING DO YOU DESIRE ?

 (3MAX)

 ENTER NUMBER OF LINES.

 (ENTER "0" FOR NO HEADING)

3

 A MAXIMUM OF 32 CHARACTERS PER LINE IS

 ALLCWED

 LINE 1 IS:

aircraft with an autopilot in

LINE 2 IS:
the longitudinal mode

LINE 3 IS:
ref. 7; page 315

PLOT HEADING

LINE 1 : AIRCRAFT WITH AN AUTOPILOT IN
LINE 2 : THE LONGITUDINAL MODE
LINE 3 : REF. 7; PAGE 315

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

>USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

>USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

A DISPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

POLE-ZERO MAP
OPEN LOOP TRANSFER FUNCTION
DO YOU DESIRE TO MAKE ANY CHANGES TO:

1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS
3. HEADING
4. NO ADDITIONAL CHANGES - PLOT POLE-ZERO
MAP
5. NO CHANGES - EXIT POLE-ZERO PLOTTING
ROUTINE

ENTER OPTION NUMBER.

5

POLE-ZERO MAP
OPEN LOOP TRANSFER FUNCTION
DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

3

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

CPTGRAPH

ENTER OPTICN NUMBER.

7

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. CPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTICN NUMBER.

4

```
R; T=2.80/4.67 19:45:15
record off
END RECORDING OF TERMINAL SESSION
```

The preceding example follows at the end of the chapter as figure 3.1 .

b. Example of Tabular Data (Terminal Session)

The entering and exiting option menus for the tabular data terminal session are identical to the graphic response terminal session option menus, and have been eliminated from the following terminal session.

```
BEGIN RECORDING OF TERMINAL SESSION
R; T=0.01/0.02 19:51:03
EXECUTION BEGINS...
```

```
POLE-ZERO MAP
OPEN LOCP TRANSFER FUNCTION
DO YOU DESIRE:
    1. GRAPHICAL OUTPUT
    2. TABULAR DATA
    3. QUIT SUBROUTINE
ENTER OPTICN NUMBER.
2
```

```
POLE-ZERO MAP
OPEN LOCP TRANSFER FUNCTION
DO YOU DESIRE TABULAR OUTPUT TO GO TO:
    1. SCREEN
    2. PRINTER
    3. DISK (OPGRAPH LISTING)
ENTER OPTICN NUMBER.
1
```


POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

INPUT NUMBER = 1

OUTPUT NUMBER = 1

TRANSFER FUNCTION (DC) GAIN = 0.1000D+01

DENOMINATOR ORDER = 4

POLE LOCATIONS

REAL PART	IMAGINARY PART
0.0000D+00	0.0000D+00
0.1000D+01	0.0000D+00
-0.2000D+01	0.3464D+01
-0.2000D+01	-0.3464D+01

NUMERATOR ORDER = 1

ZERO LOCATIONS

REAL PART	IMAGINARY PART
-0.1000D+01	0.0000D+00

DO YOU DESIRE TO CHANGE OUTPUT DEVICE?

TYPE "YES" OR "NC".

n

POLE-ZERO MAP

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

3

R; T=1.60/3.23 19:53:00

record off

3. Root-Locus

The subroutine RTLO creates a closed loop system (with negative unity feed-back) from the selected transfer function input numbers and output numbers. Transfer function poles and zeros are formed into two separate pclynomials by the subroutine MAKPOL. The numerator polynomial (zeros) is multiplied by the gains, and the two polynomials are added to form a single polynomial. Complex roots of the of the pclynomial are calculated by the IMSL library subroutine ZRPCLY. The ccmplex roots are separated into real and imaginary components for either plotting or tabulating.

The plotting routine first plots the transfer function pole and zero locations then plots the root locations as they are received from the subroutine ZRPOLY. For the plotting routine the gain interval specified by the user is divided into 2000 evenly spaced points for the subroutine ZRPOLY to calculate roots. The poles, zeros, and roots are plotted within the plct limits specified by the user.

For the tabular data routine the user is given the option to choose between 1 and 500 points for the subroutine ZRPOLY to calculate roots. The tabular data output provided for the user is system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, the pole and zero locations, and the root locus gains and roots

A positive unity feedback system may be analyzed by specifying a negative gain range vice a positive gain range.

4. Root-Locus (Interactive Example)

The following interactive example is of the open loop transfer function for an aircraft with an autopilot in the longitudinal mode described in Pole-Zero interactive section.

a. Example of Graphic Response (Terminal Session)

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 20:28:37

EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

1

OPTGRAPH

THE OPEN LOOP TRANSFER SYSTEM OBTAINED FROM OPTSYS
CONTAINS:

1. NUMBER OF STATES = 4
2. NUMBER OF CONTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,

AND/OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

CUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
 2. FOCT-LOCUS
 3. BODE
 4. NYQUIST (POLAR PLOT)
 5. NICHOLS
 6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
 7. ANOTHER TYPE TRANSFER FUNCTION /EXIT
- OPTGRAPH

ENTER OPTION NUMBER.

2

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

1

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

PLOTTER SELECTION AND PAGE SIZE

(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618

2. VERSATEC

CHCCSE OPTION 1 CR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

6

WIDTH =

8

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = ?

-15

X MAXIMUM = ?

5

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = ?

-5

Y MAXIMUM = ?

5

PLOTTING LIMITS FOR GRAPH

X AXIS (REAL AXIS)

X MINIMUM = -15.00

X MAXIMUM = 5.00

Y AXIS (IMAGINARY AXIS)

Y MINIMUM = -5.00

Y MAXIMUM = 5.00

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = ?

0

MAXIMUM GAIN = ?

500

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = 0.000D+00

MAXIMUM GAIN = 0.500D+03

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

PLOT HEADING

HOW MANY LINES OF HEADING DO YOU DESIRE ?

(3 MAX)

ENTER NUMBER OF LINES.

(ENTER "0" FOR NO HEADING)

3

A MAXIMUM OF 32 CHARACTERS PER LINE IS
ALLOWED

LINE 1 IS:

aircraft with an autopilot in

LINE 2 IS:

the longitudinal mode

LINE 3 IS:

ref. 7; page 315

PLOT HEADING

LINE 1 : AIRCRAFT WITH AN AUTOPILOT IN
LINE 2 : THE LONGITUDINAL MODE
LINE 3 : REF. 7; PAGE 315

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

>USING A PRE-ALLOCATED DATASET FOR UNIT FT17F001.

>USING A PRE-ALLOCATED DATASET FOR UNIT FT18F001.

A DISSPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TO MAKE ANY CHANGES TO:

1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS
3. HEADING
4. GAIN RANGE
5. NO ADDITIONAL CHANGES - PLOT ROOT-LOCUS
6. NO CHANGES - EXIT ROOT-LOCUS PLOTTING

ROUTINE

ENTER OPTICN NUMBER.

6

RCOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

3

OPTGRAPH

OPEN LOOP TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

OPTGRAPH

ENTER OPTION NUMBER.

7

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

4

R; T=12.36/15.14 20:32:54

record off

END RECORDING OF TERMINAL SESSION

The preceding example follows at the end of the chapter as figure 3.2 .

b. Example of tabular data (Terminal Session)

The entering and exiting option menus for the tabular data terminal session are identical to the graphic response terminal session option menus, and have been eliminated from the following terminal session.

BEGIN RECORDING OF TERMINAL SESSION

R; T=0.01/0.02 20:42:59

EXECUTION BEGINS...

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTICN NUMBER.

2

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TABULAR OUTPUT TO GO TO:

1. SCREEN
2. PRINTER
3. DISK (OPGRAPH LISTING)

ENTER OPTICN NUMBER.

1

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = ?

1

MAXIMUM GAIN = ?

5

GAIN RANGE FOR ROOT-LOCUS PLOT

MINIMUM GAIN = 0.100D+01

MAXIMUM GAIN = 0.500D+01

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NC".

n

HOW MANY PCINTS DO YOU WANT TO TABULATE?

(500 IS THE MAXIMUM)

ENTER NUMBER OF POINTS.

5

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

INPUT NUMBER = 1

CUTPUT NUMBER = 1

TRANSFER FUNCTION (DC) GAIN = 0.1000D+01

OPEN LOOP TF POLES AND ZEROS

DENOMINATOR ORDER = 4

POLE LOCATIONS

REAL PART	IMAGINARY PART
0.0000D+00	0.0000D+00
0.1000D+01	0.0000D+00
-0.2000D+01	0.3464D+01
-0.2000D+01	-0.3464D+01

NUMERATOR ORDER = 1

ZERO LOCATIONS

REAL PART	IMAGINARY PART
-0.1000D+01	0.0000D+00

GAIN = 0.1000D+01

REAL PART	IMAGINARY PART
0.7074D-01	0.0000D+00
0.8962D+00	0.0000D+00
-0.1983D+01	0.3441D+01

-0.1983D+01 -0.3441D+01

GAIN = 0.1800D+01

REAL PART	IMAGINARY PART
0.1453D+00	0.0000D+00
0.7948D+00	0.0000D+00
-0.1970D+01	0.3422D+01
-0.1970D+01	-0.3422D+01

GAIN = 0.2600D+01

REAL PART	IMAGINARY PART
0.2576D+00	0.0000D+00
0.6552D+00	0.0000D+00
-0.1956D+01	0.3402D+01
-0.1956D+01	-0.3402D+01

GAIN = 0.3400D+01

REAL PART	IMAGINARY PART
0.4426D+00	0.1661D+00
0.4426D+00	-0.1661D+00
-0.1943D+01	0.3383D+01
-0.1943D+01	-0.3383D+01

GAIN = 0.4200D+01

REAL PART	IMAGINARY PART
0.4285D+00	0.3096D+00
0.4285D+00	-0.3096D+00
-0.1929D+01	0.3363D+01
-0.1929D+01	-0.3363D+01

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

Y

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE TO MAKE ANY CHANGES TO:

1. OUTPUT DEVICE

2. GAIN RANGE
3. NUMBER OF POINTS CALCULATED
4. NO ADDITIONAL CHANGES - TABULATE DATA
5. EXIT ROOT-LCCUS TABULAR DATA ROUTINE

ENTER CPTICN NUMBER.

5

ROOT-LOCUS

OPEN LOOP TRANSFER FUNCTION

DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER CPTICN NUMBER.

3

R; T=1.63/3.40 20:44:54

record off

END RECORDING OF TERMINAL SESSION

G. FREQUENCY RESPONSE ANALYSIS

The calculations for the frequency response of the Bode, Nyquist, and Nichols responses are identical. The only difference is in the graphical presentation of the calculated data. The subroutines BODE, NYQST, and NICHOLS call the subroutine FREQ to calculate the transfer function magnitude and phase for the frequency range specified by the user.

The subroutine FREQ eliminates the extraneous zeros by comparing the zeros to the Markov parameter as described in the Characteristic Equation Analysis section. The frequency response magnitude and phase are calculated using double precision complex library functions following the

conventions of the IBM System /360/370/ FORTRAN IV language. The frequency response is calculated by determining the product of successive numerator zeros which have been combined with the frequency and divided by the product of successive denominator poles which also have been combined with the frequency. The magnitude for the complex result of the zeros, poles, and frequency calculations is determined using the FORTRAN double precision library functions for finding the magnitude of a complex argument. The phase of the of the complex result is determined by separating the real and imaginary parts of the complex results, and finding the phase angle using the double precision FORTRAN library function arctangent. The angle is converted from radians to degrees and is corrected for proper quadrant by checking the signs of the real and imaginary parts. The normal range for calculating phase angles by the subroutine is from +540 degrees to -540 degrees. This range is considered adequate for most application, but should these limits be exceeded the plot will jump with a vertical line to either +360 degrees or -360 degrees respectively before continuing with the plot.

For a graphic response the user specified frequency range is divided into 500 equally spaced points (on a logarithmic scale) for which to calculate the magnitude and phase. The user is given the option to select between 1 and 500 points for tabular data.

1. Bode Response

The graphic response section of the subroutine BODE creates two plots. The first plot is the response magnitude in decibels versus the frequency in radians per second. The magnitude from the subroutine FREQ is converted to decibels and plotted with its corresponding frequency. The second plot is of the phase in degrees versus frequency in radians

per second. The phase from the subroutine FREQ is plotted with its corresponding frequency. The frequency for both plots is plotted on a logarithmic scale. The scale range for both is set automatically to include the entire range of data to be plotted by the subroutine MINMAX which scans the data for the maximum and minimum values. System information (transfer function input number, output number, and (DC) gain) is listed in the graph legend.

The magnitude in decibels and the phase in degrees is searched for "0" decibel and 180 degree crossover points to calculate the phase and gain margins.

The tabular response section gives the user the option to select the the number of frequency points to be calculated between 1 and 500 for the frequency range he has specified. For tabular data response the user is given system information (numerator order, denominator order, and transfer function (DC) gain) for the selected transfer function input number and output number, and the frequency (radians per second), magnitude (decibels), phase (degrees), and the real and imaginary frequency response parts.

2. Bode Response (Interactive Example)

The following example of compensator transfer function with a filter and regulator was obtained from [Ref. 8, pp. 382 - 384].

The compensated system is represented by:

$$[C] = [-254.1 \quad -19.57]$$

$$[D] = [0]$$

$$[F] = \begin{bmatrix} 0. & 1.0 \\ 0. & -4.6 \end{bmatrix}$$

[G] = | 0. |
 | 0.787 |

[GAM] = | 0. |
 | 0.1 |

[H] = [1. 0.]

[K] = | 95.4 |
 | 4561. |

[Q] = [10.]

a. Example of Graphic Response (Terminal Session)

BEGIN RECCRDING OF TERMINAL SESSION
R; T=0.01/0.02 18:30:23
graphics
EXECUTION BEGINS...

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

3

OPTGRAPH

THE COMPENSATOR TRANSFER SYSTEM OBTAINED FROM
OPTSYS CCNTAINS:

1. NUMBER OF STATES = 2
2. NUMBER OF CONTROLS (INPUTS) = 1
3. NUMBER OF MEASUREMENTS (OUTPUTS) = 1
4. MARKOV PARAMETER = 10.0×10^{-6}

CLEAR SCREEN TO CONTINUE

OPTGRAPH

AN OPEN LOOP POLE-ZERO, ROOT LOCUS, BODE, NYQUIST,
AND/OR NICHOLS PLOT IS DESIRED FOR:

INPUT # ?

1

OUTPUT # ?

1

OPTGRAPH

COMPENSATOR TRANSFER FUNCTION

INPUT # = 1

OUTPUT # = 1

DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:

1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME
TRANSFER FUNCTION)
7. ANOTHER TYPE TRANSFER FUNCTION /EXIT

CPTGRAPH

ENTER OPTION NUMBER.

3

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE:

1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE

ENTER OPTION NUMBER.

1

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
PLOTTER SELECTION AND PAGE SIZE
(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH
SELECTION OF TEK618)

WHICH PLOTTER DO YOU DESIRE:

1. TEK618
2. VERSATEC

CHOOSE OPTION 1 OR 2

2

PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0
INCHES)

HEIGHT =

5.65

WIDTH =

8.5

PLOTTING LIMITS FOR GRAPH

FREQUENCY RANGE - RADIANS PER SECOND

MINIMUM FREQUENCY = ?

0.1

MAXIMUM FREQUENCY = ?

1000000

PLOTTING LIMITS FOR GRAPH

FREQUENCY RANGE - RADIANS PER SECOND

MINIMUM FREQUENCY = 0.100E+00
MAXIMUM FREQUENCY = 0.100E+07
DO YOU DESIRE TO MAKE ANY CHANGES ?
TYPE "YES" OR "NO".

n

PLOT HEADING

HOW MANY LINES OF HEADING DO YOU DESIRE ?
(3 MAX)

ENTER NUMBER OF LINES.

(ENTER "0" FOR NO HEADING)

2

A MAXIMUM OF 32 CHARACTERS PER LINE IS
ALLOWED

LINE 1 IS:

filter simulation

LINE 2 IS:

ref. 8; pages 332 - 334

PLOT HEADING

LINE 1 : FILTER SIMULATION

LINE 2 : REF. 8; PAGES 332 - 334

DO YOU DESIRE TO MAKE ANY CHANGES ?

TYPE "YES" OR "NO".

n

A DISSPLA METAFILE HAS BEEN CREATED
CLEAR SCREEN TO CONTINUE

GAIN MARGIN AND PHASE MARGIN
PHASE CROSSOVER FREQUENCY IS 75.1184 RAD/SEC;
GAIN MARGIN IS -59.8906 DB
GAIN CROSSOVER FREQUENCY IS ***** RAD/SEC;
PHASE MARGIN IS -89.9480 DEG
CLEAR SCREEN TO CONTINUE

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE TO MAKE ANY CHANGES TO:
1. PLOTTER / PAGE SIZE
2. GRAPH LIMITS (FREQUENCY RANGE)
3. HEADING
4. NO ADDITIONAL CHANGES - PLOT BODE PLOTS
5. NO CHANGES - EXIT BODE SUBROUTINE
ENTER OPTION NUMBER.
5

BODE PLOT
COMPENSATOR TRANSFER FUNCTION
DO YOU DESIRE:
1. GRAPHICAL OUTPUT
2. TABULAR DATA
3. QUIT SUBROUTINE
ENTER OPTION NUMBER.
3

OPTGRAPH
COMPENSATOR TRANSFER FUNCTION
INPUT # = 1
OUTPUT # = 1
DO YOU DESIRE GRAPHICAL RESPONSE AND/OR TABULAR
DATA FOR:
1. POLE-ZERO MAP
2. ROOT-LOCUS
3. BODE
4. NYQUIST (POLAR PLOT)
5. NICHOLS
6. ANOTHER INPUT/OUTPUT COMBINATION (SAME

TRANSFER FUNCTION)

7. ANOTHER TYPE TRANSFER FUNCTION /EXIT
CPTGRAPH

ENTER OPTION NUMBER.

7

OPTGRAPH

DO YOU DESIRE TO ANALYZE:

1. OPEN LOOP TRANSFER FUNCTION
2. NOISE TRANSFER FUNCTION
3. COMPENSATOR TRANSFER FUNCTION
4. EXIT OPTGRAPH

ENTER OPTION NUMBER.

4

R; T=7.01/8.84 18:39:11

record off

END RECORDING OF TERMINAL SESSION

The preceding example follows at the end of the chapter as figure 3.3 and figure 3.4 .

b. Example of Tabular Data Listing File

The terminal session for the Bode tabular data response is similar to the Bode graphic response terminal session. Appendix D contains an example of tabular output sent to a disk (OPGRAPH LISTING) for five points between 10 and 100 radians/second.

3. Nyquist Response

The graphic response section of the subroutine NYQST creates a polar plot from the magnitude and phase data received from the subroutine FREQ. The plot is automatically scaled from a minimum radius of one by using the

subroutine MINMAX to determine the maximum phase and magnitude values. The gain / phase margin calculations and the tabular data output are identical to the subroutine BODE.

4. Nyquist Response (Interactive Example)

The following example is of the open loop transfer function for the filter simulation example described in the Bode interactive section.

This example follows at the end of the chapter as figure 3.5 .

5. Nichols Response

The graphic response section of the subroutine NICHCI creates a single rectangular plot of magnitude in decibels versus phase in degrees. The magnitude from the subroutine FREQ is converted to decibels and plotted with the corresponding magnitude. The plot is automatically scaled using the subroutine MINMAX to determine the maximum phase and magnitude values. The gain / phase margin calculations and the tabular data output are identical to the subroutine BODE.

6. Nichols Response (Interactive Example)

The following example is of the open loop transfer function for the filter simulation example described in the Bode interactive section.

This example follows at the end of the chapter as figure 3.6 .

AIRCRAFT WITH AN AUTOPILOT IN
THE LONGITUDINAL MODE
REF. 7; PAGE 315
OPEN LOOP TRANSFER POLE-ZERO MAP

INPUT # = 1
OUTPUT # = 1
DC GAIN = 1.000×10^0

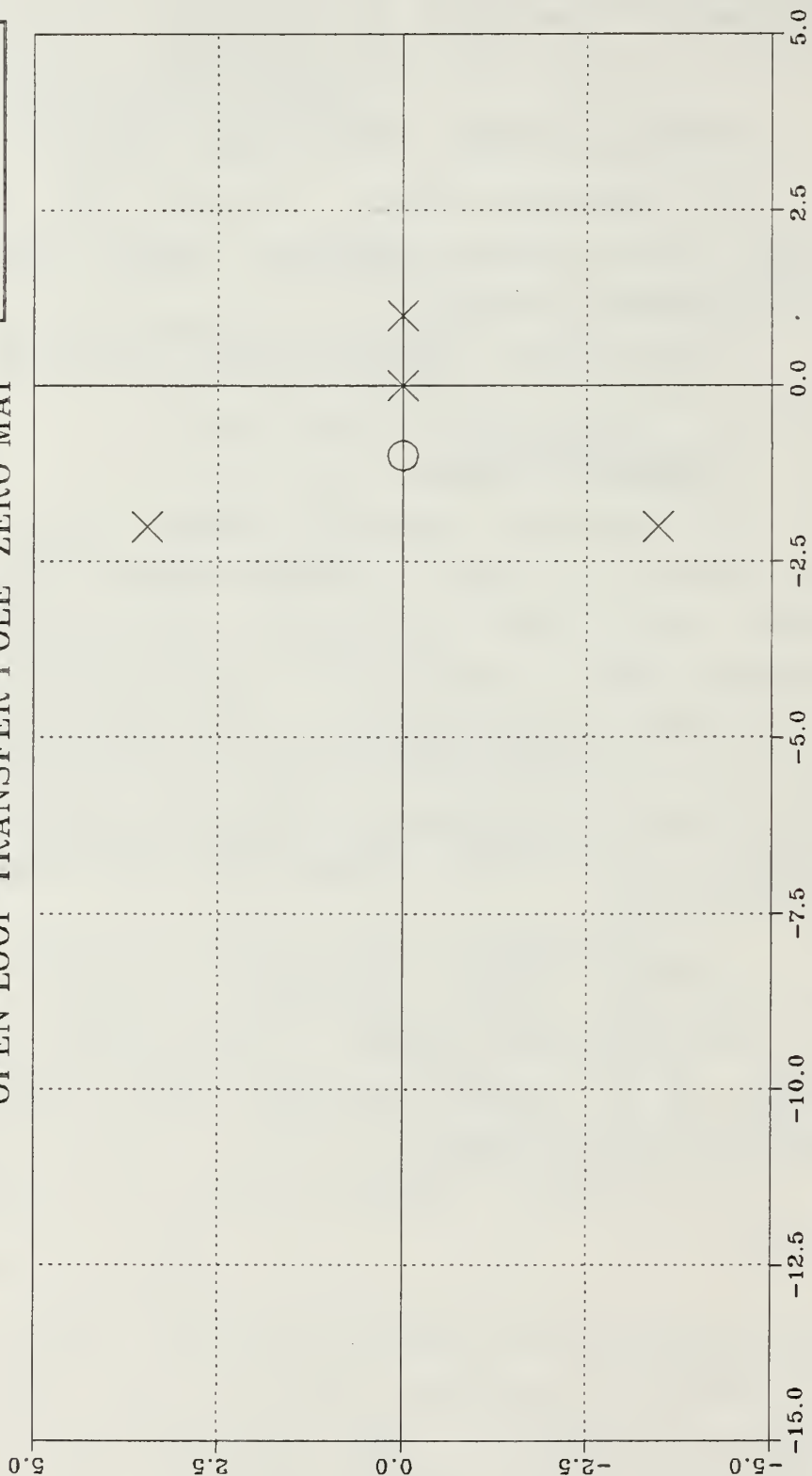


Figure 3.1 Pole-Zero Map Example.

AIRCRAFT WITH AN AUTO PILOT IN
THE LONGITUDINAL MODE
REF. 7; PAGE 315
ROOT-LOCUS PLOT (OPEN LOOP TF)

INPUT # = 1
OUTPUT # = 1
DC GAIN = $1,000 \times 10^0$

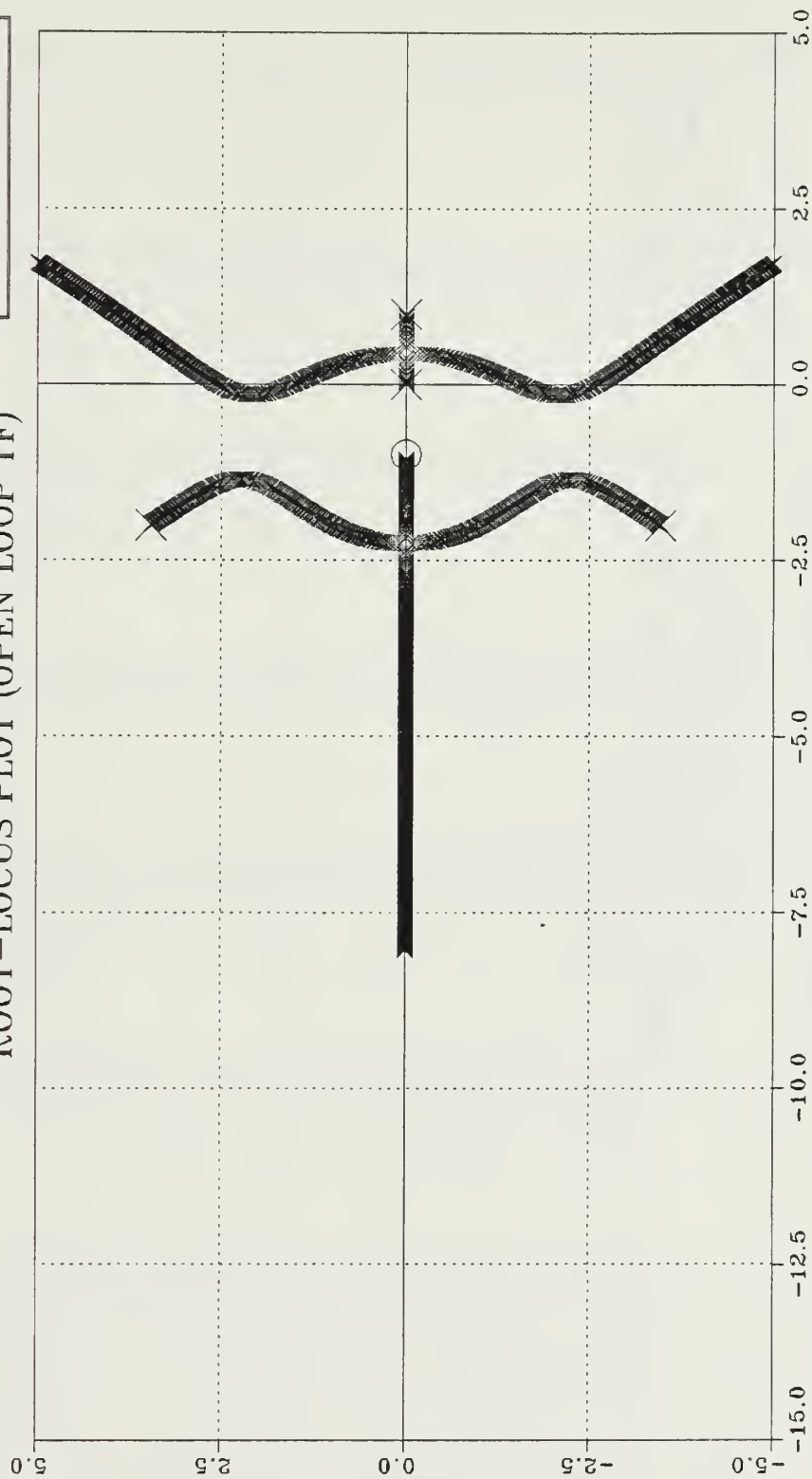


Figure 3.2 Root Locus Response Example.

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 COMPENSATOR TF BODE MAGNITUDE

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = -1.135×10^5

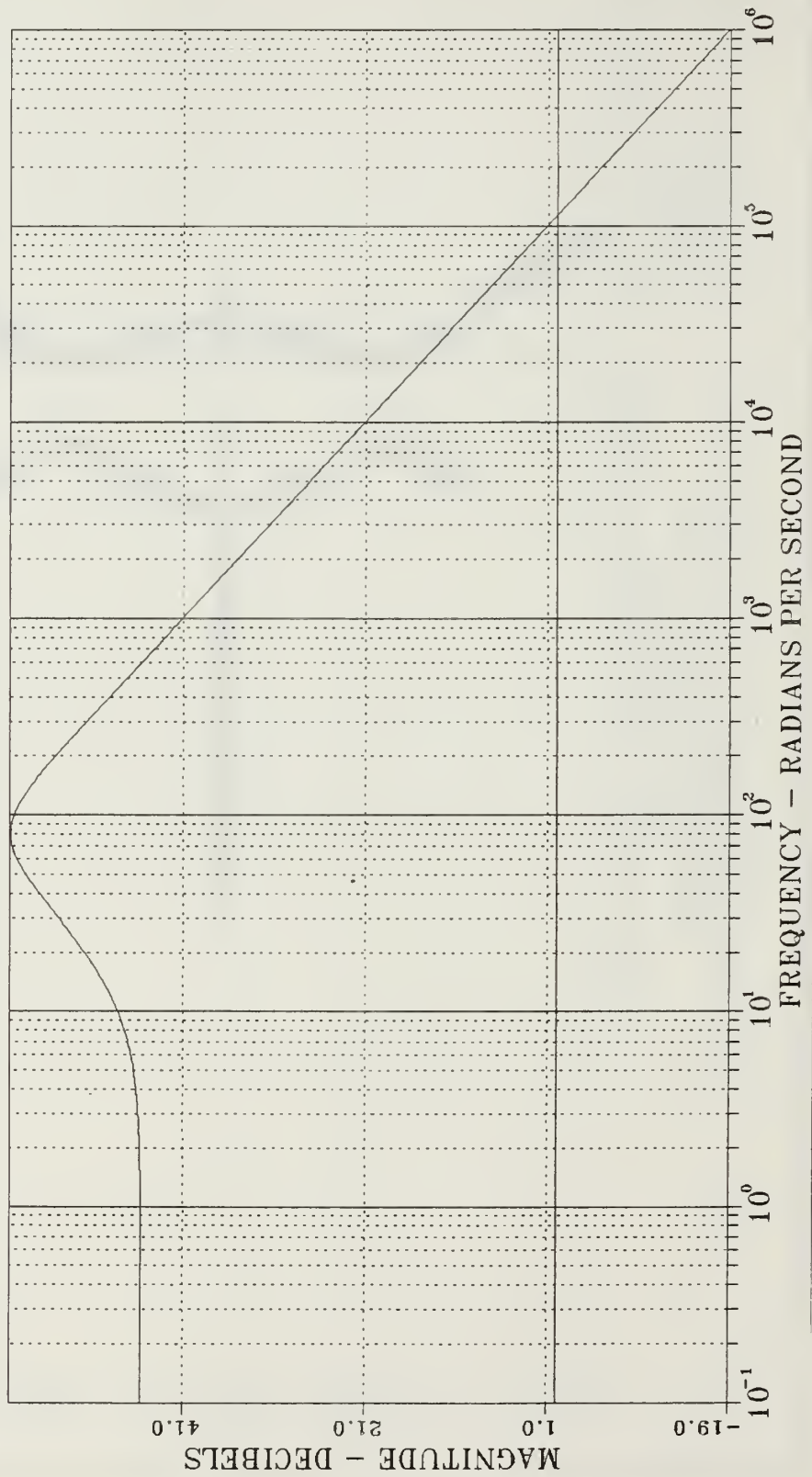


Figure 3.3 Bode Response Example (Magnitude).

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 COMPENSATOR TF BODE PHASE

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = -1.135×10^0

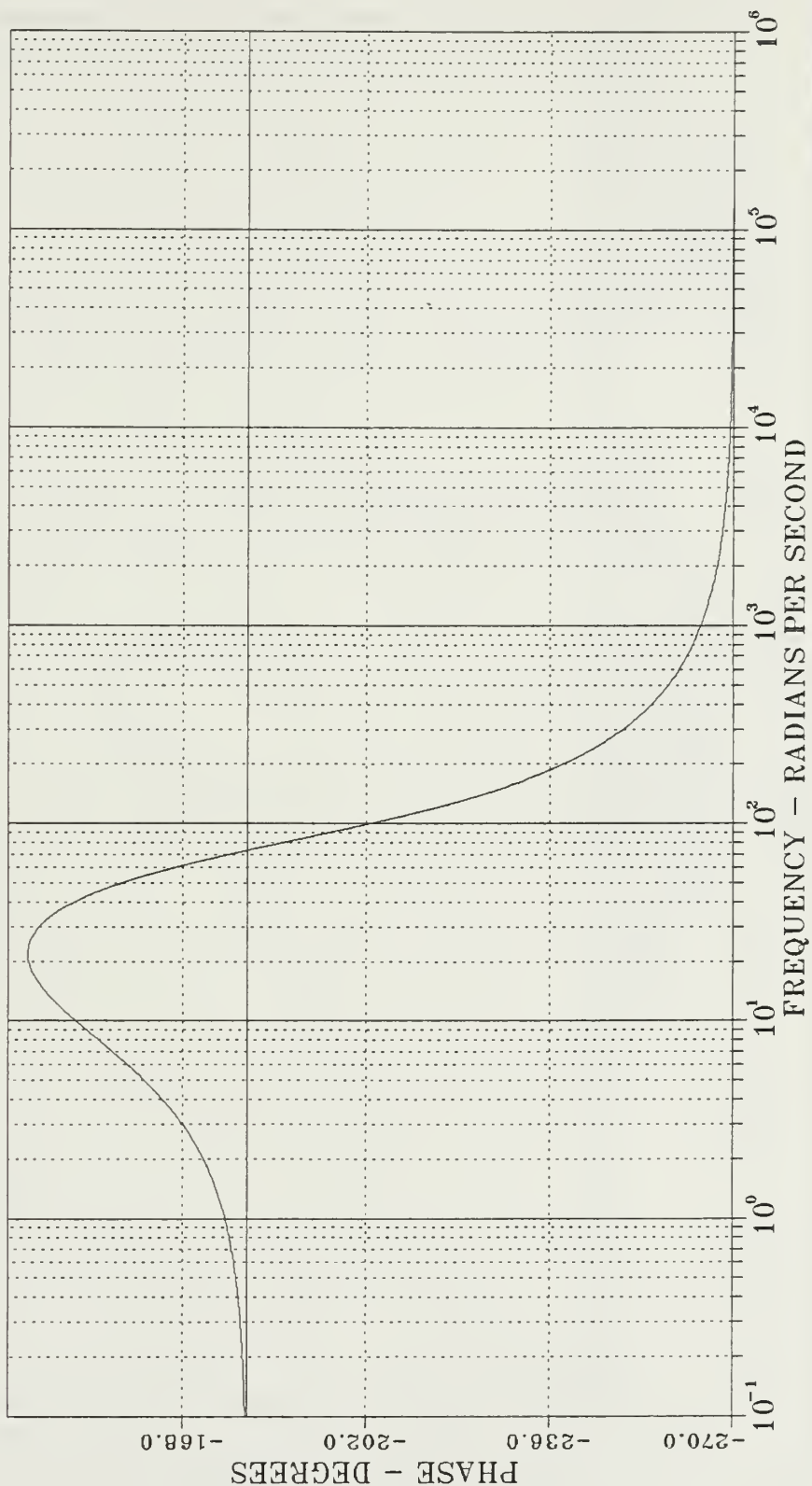


Figure 3.4 Bode Response Example (Phase).

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 OPEN LOOP TRANSFER NYQUIST

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = 7.870×10^{-1}

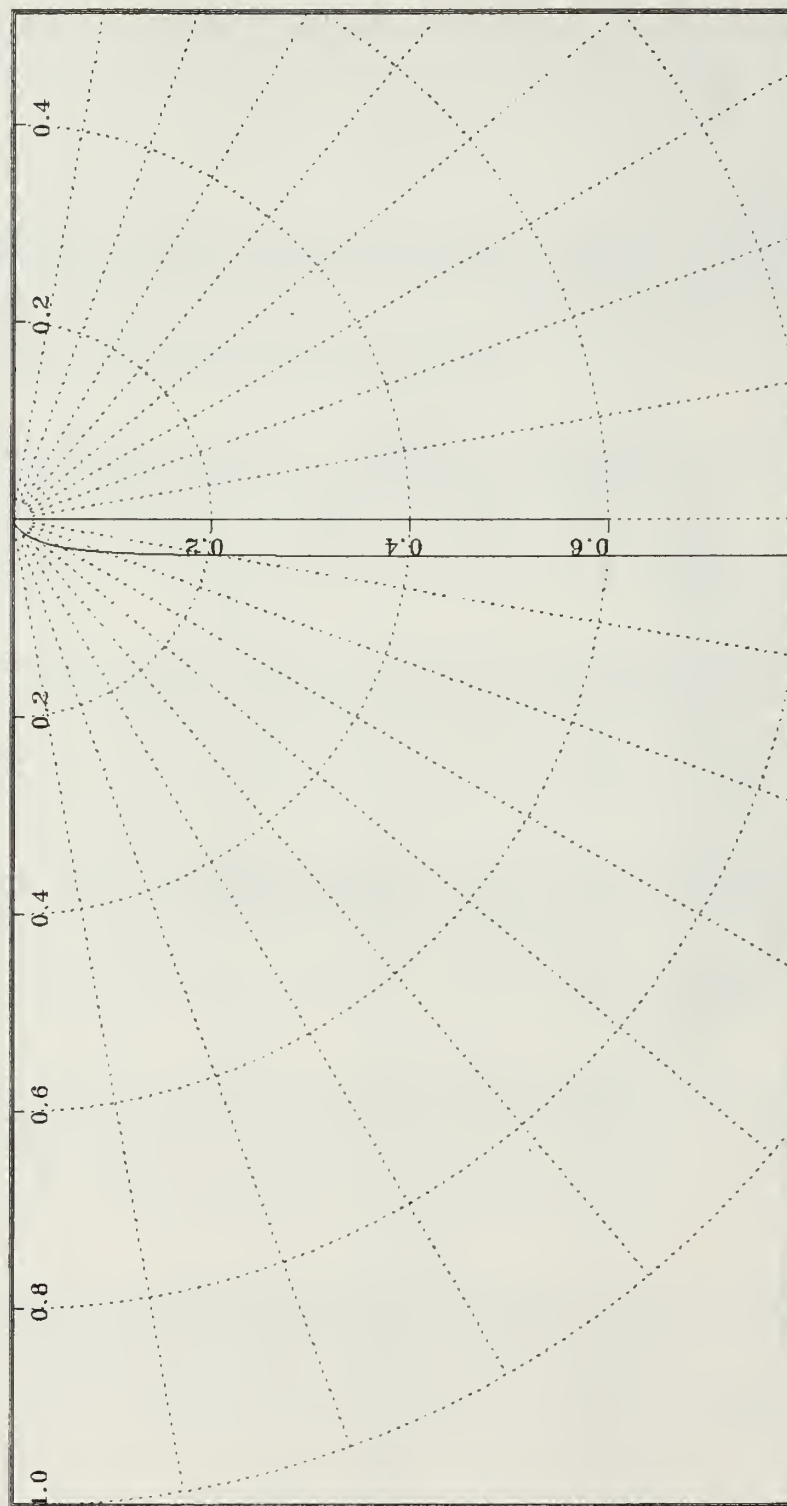


Figure 3.5 Nyquist Response Example.

FILTER SIMULATION
 REF. 8; PAGES 332 - 334
 OPEN LOOP TRANSFER NICHOLS

INPUT # = 1
 OUTPUT # = 1
 DC GAIN = 7.870×10^{-1}

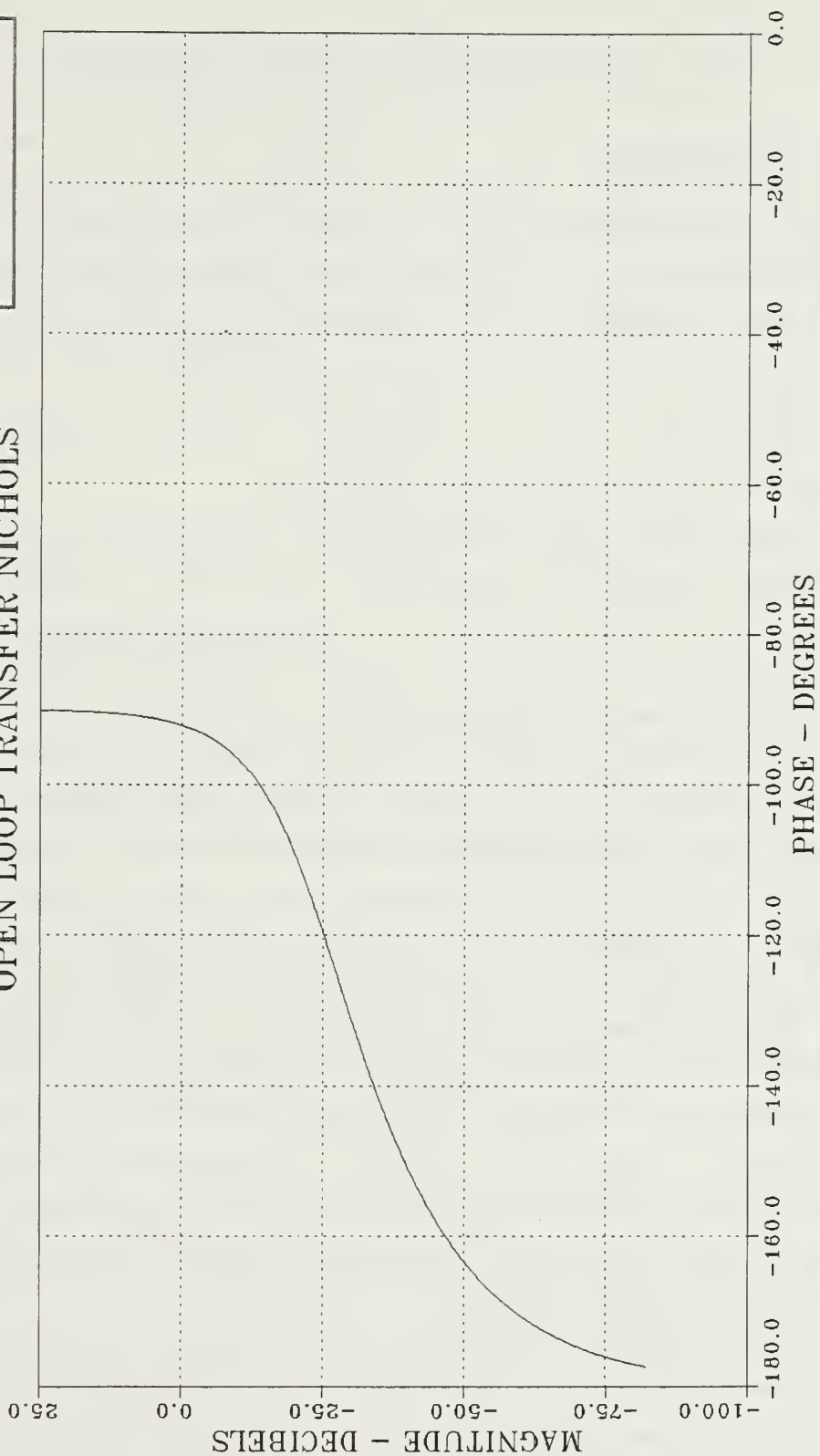


Figure 3.6 Nichols Response Example.

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

An evaluation of the computational ability of OPTSYSX and OPTGRAPH, the program was tested using an 82 X 82 matrix, provided by NASA-Edwards, of longitudinal equations for the X-29A experimental forward-swept wing fighter aircraft prototype. The OPTSYSX program array had to be redimensioned and a 2-megabyte virtual machine size was required for a system of this size.

The open loop eigenvalues calculated by the OPTSYSX program compared favorably with the Eigensystem data supplied by NASA-Edwards, but the OPTSYSX program failed in calculating the open loop transfer function zeros. Extraneous zeros (calculated by OPTSYSX) were of the same approximate magnitude as the open loop system zeros. This prevented the identification and elimination of the extraneous zeros by comparison with the Markov parameter. The open loop zero identification was further complicated by not having available open loop transfer function zero information from NASA-Edwards to compare the OPTSYSX results to.

When requested for additional information NASA-Edwards provided a computer tape with a revised 98 X 98 matrix of longitudinal motion equations for the X-29A fighter aircraft. For this revised system, the OPTSYSX program requires a virtual machine size in excess of 2-megabyte. Time constraints have prevented running the OPTSYSX program with the revised system.

The OPTGRAPH program was tested using the OPTSYSX program's open loop Eigenvalue and transfer function output data for the X-29A 82 X 82 system of longitudinal motion

equations. OPTGRAPH provided excellent quality Bode plots for all of the open loop transfer function inputs and outputs and did not exhibit any computational difficulties as indicated by the lack of error messages at the end of the test. While the test results are encouraging as to the ability of the OPTGRAPH program to assist in the analysis of large order systems, the actual computational accuracy of the OPTGRAPH program can not be fully verified until accurate transfer function information can be obtained from the OPTSYSX program.

B. RECCMMENDATIONS

Based on the results of this Thesis the following areas emerged as possible areas for further research and study.

1. Root Finding Routines

The OPTSYSX program has two similar root finding subroutines (HQR and HQR2), to find the transfer function Eigensystem (poles) and zeros. The order of the transfer function numerator is calculated separately in the subroutine ZEROS.

The inclusion of the open loop transfer function eigensystem (pole locations), zero locations, transfer function numerator orders, and transfer function gains, for the revised (98 X 98) longitudinal mode system by NASA-Edwards should simplify the locating the problem in calculating the transfer function zeros.

The open loop system data from NASA-Edwards also included Bode magnitude and phase plots, and gain and phase margin information, which can be used to evaluate more conclusively the OPTGRAPH program's ability to help evaluate a large order system.

2. Program Memory Requirements

The OPTSYSX program requires in excess of 2-megabyte of virtual machine memory when configured for large matrix operations (98 X 98). Virtual machines with this memory capacity are not normally available to a user. The memory usage for matrix storage and manipulation is a possible area for the reduction in program memory requirement size. The OPTSYSX program retains most of its computational arrays in memory. This method simplifies programming but is extremely inefficient in its use of virtual machine memory.

3. Further Modifications

The area of modern digital controls should be further investigated. The computational abilities of the OPTGRAPH program make it readily adaptable, with minor modifications, for assisting in the analysis of transfer functions in the digital domains.

APPENDIX A
OPTSYS EXEC LISTING

ETRACE OFF

```
*
*           THE OPTSYS EXEC
*           CONTROLS THE OPTSYSX, OPTCALC AND OPTPLOT
*           TO DETERMINE THE TIME RESPONSE OF A SYSTEM.
*           BY H. A. DIEL
*           VERSION 1.0    16 JULY 1984
*           MODIFIED BY M. H. LAPTAS
```

CLRSCRN

EBEGTYPE -ENDFOUR

```
      CUIPUT FROM OPTSYSX WILL COME TO THE SCREEN IF YOU
      WISH DISK FILE
      ENTER  D      ANY OTHER INPUT YIELD SCREEN
```

-ENDFOUR

ERead VARS EANS

EIF .EANS NE .D EGOTO -ONE

FILEDEF 06 DISK OUTPUTX LISTING A1

CLRSCRN

EBEGTYPE -ENDSIX

OUTPUT WILL GO TO DISK FILE 'OUTPUTX LISTING A1'

-ENDSIX

CP SLEEP 3 SEC

-ONE

CLRSCRN

CP LINK 0039P 191 251 RR

ACC 251 F

* *

* ALLOW THE USE OF AN OLD "OPTMAT DATA A1" *

* *

RENAME OPTMAT DATA A1 OPTSYS DATA A1

EIF &RC NE 0 &GOTO -FIRST

RENAME OPTSYS DATA A1 OPTMAT DATA A1

FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1

EREAD VARS &A1 &A2

EIF .&A2 NE .1 &GOTO -START

CLRSCRN

EBEGTYPE -ENDTWO

YOU HAVE A DATA FILE NAMED 'OPTMAT DATA' ON YOUR A
DISK THAT WAS PREVIOUSLY GENERATED BY THE OPTSYS
PROGRAM AND CONTAINS THE F, G, H, GAMMA, A AND B
MATRICES FROM THAT RUN.

IF YOU WOULD LIKE TO USE THESE SAME MATRICES FOR
THIS RUN, THE CPTSYS PROGRAM WILL READ IN THE
DESIRED DATA AT THE APPROPRIATE TIME,

IF YOU TYPE (Y) ES.

ANY OTHER INPUT WILL RESULT IN THAT FILE BEING ERASED!

-ENDTWO

&READ VARS &ANS

&IF .&ANS EQ .Y &GOTO -START

-FIRST

* *

* ERASE THE OLD "OPTMAT DATA A1" DATA FILE *

* PLACE "000 0" IN THE NEW "OPTMAT DATA FILE" *

* TO ACT AS A FLAG FOR OPTSYSX AND OPTCALC *

* *

ERASE OPTMAT DATA A1

&STACK 000 0

FILESTCK OPTMAT DATA A1 F 80 1

-START

CLRSCFN

&BEGTYPE -ENDONE

THE OPTSYS EXEC CONTROLS FOUR PROGRAMS:

- 1 OPTSYSX FORTRAN (SYSTEM ANALYSIS)
GENERATES OPTMAT AND OPTGRAPH DATA SETS
- RERUN USES OPTMAT
- 2 OPTCALC FORTRAN (CALCULATE TIME RESPONSE)
GENERATES OPTPLOT DATA SET USES OPTMAT FROM
OPTSYSX
- 3 OPTGRAPH FORTRAN (POLE-ZERO, ROOT-LOCUS, BODE,

NYQUIST, ETC)

USES OPTGROL -OPTGRNO -OPTGRCM FROM OPTSYSX

4 EXIT

OPTPLOT FORTRAN IS THE FOURTH PROGRAM

USES OPTPLOT DATA SET FROM OPTCALC

THE SIZE OF THE DATA SETS VARY WITH THE SYSTEM
ORDER, AND CAN USE ABOUT 20% OF THE USERS DISK SPACE.
THEREFORE ENSURE THAT SUFFICIENT DISK SPACE IS
AVAILABLE.

IF DATA SET IS ALREADY AVAILABLE YOU MAY RUN ANY
PROGRAM

- ENTER 1 , 2 , 3 , 4 ANY OTHER INPUT
RETURNS TO MENU

-ENDONE

&READ VARS &ANS

&IF .&ANS EQ .1 &GOTO -OPTSYS

&IF .&ANS EQ .2 &GOTO -OPTCALC1

&IF .&ANS EQ .3 &GOTO -OPGRAPH1

&IF .&ANS EQ .4 &EXIT &RC

&GOTO -START

-OPTSYS

FILEDEF 8 DISK OPTPLCT DATA A1 (PERM

FILEDEF 9 DISK OPTMAT DATA A1 (PERM

FILEDEF 10 DISK OPTGROL DATA A1 (PERM

FILEDEF 01 DISK OPTGRNO DATA A1 (PERM

FILEDEF 04 DISK OPTGRCM DATA A1 (PERM

GLOBAL TXTLIB VFORTLIB IMSLDP NONIMSL

ETYPE LOADING OPTSYS..... GENERAL STATE VARIABLE ANALYSIS

PROGRAM

ETYPE..... OPTMAT DATA A1 INPUT DATA SET ON RERUNS
OPTSYSX

CLRSCRN

ETYPE... DATA SETS .. OPTMAT AND .. OPGRAPH CREATED

CP SLEEP 5 SEC

&GOTO -START

-OPTCALC1

* *

* CHECK FOR DATA IN THE FILE "OPTMAT DATA "

* BEFORE LOADING OPTCALC *

* *

FINDSTAK OPTMAT DATA A1 C01 0 LIM002 ALL GROUP1

&READ VARS &A1 &A2

&IF .&A2 EQ .1 &GOTO -OPTCALC

&TYPE PROPER DATA FILE IS NOT AVAILABLE FOR OPTCALC

CP SLEEP 5 SEC

&GOTO -START

-OPTCALC

&TYPE.....OPTCALC.... TIME RESPONSE PROGRAM

FILEDEF 8 DISK OPTPLCT DATA A1 (PERM

FILEDEF 9 DISK OPTMAT DATA A1 (PERM

GLOBAL TXTLIB VFORTLIE IMSLDP NONIMSL

OPTCAIC

&TYPE .. OPTPLOT DATA A1 CREATED

CP SLEEP 5 SEC

CLRSCRN

&BEGTYPE -ENDNINE

IF YOU ARE DISSATISFIED WITH THE RESULTS
THUS FAR AND WOULD LIKE TO EXIT TO CMS,

-TYPE 'Y' TO EXIT-

(ANY OTHER INPUT TO CONTINUE)

```

*****
-ENDNINE
&READ VARS &ANS
&IF .&ANS EQ .Y &GOTO -START
&TYPE PLOTS NEXT
CP SLEEP 4 SEC
GLOBAL TXTLIB DISLIBVS 92DISLIB 92INTLIB VFORTLIB GRFLIB
NONIMSL CMSLIB
FILEDEF 8 DISK OPTPLOT DATA A1 (PERM
FILEDEF 11 DISK LBLANK TMP A3 (RECFM F LRECL 2400 BLKSIZE
2400 BLKSIZE 2400 XTENT 600
FILEDEF 12 DISK ISSCCMAP MAPDTA * (RECFM F LRECL 400
BLKSIZE 400
FILEDEF 13 DISK TABLET TMP A3 (RECFM VS LRECL 208 BLKSIZE
208 XTENT 1000
FILEDEF 14 DISK LBLANK MAPDTA &LBMODE (RECFM VS LRECL 608
BIKSIZE 608
FILEDEF 17 DISK DISSPIA SYSUT1 A3 (RECFM FB LRECL 2000
BLCCK 2000 XTENT 500
FILEDEF 18 DISK DISSPIA METAFILE A4 (RECFM VBS LRECL 19065
BLCCK 19069
OPTPLOT
&GOTO -META
-OPGRAPH1
*****
*
*          CHECK FOR DATA IN THE OPTGRAPH DATA SETS "
*          BEFORE LOADING OPGRAPH
*
*****
*****
RENAME OPTGROL DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -CPTNOISE
RENAME OPTSYS DATA A1 OPTGROL DATA A1

```

```

FINDSTAK OPTGROL DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-OPINCISE
RENAME OPTGRNO DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -CPTCOMP
RENAME OPTSYS DATA A1 OPTGRNO DATA A1
FINDSTAK OPTGRNO DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-OPTCCMP
RENAME OPTGRCM DATA A1 OPTSYS DATA A1
&IF &RC NE 0 &GOTO -NCDATA
RENAME OPTSYS DATA A1 OPTGRCM DATA A1
FINDSTAK OPTGRCM DATA A1 C01 0 LIM002 ALL GROUP1
&READ VARS &A1 &A2 &A3 &A4 &A5
&IF .&A2 NE .0 &GOTO -SIXTEEN
-NODATA
&PRINT PROPER DATA FILE IS NOT AVAILABLE FOR OPGRAPH
CP SLEEP 5 SEC
&GOTO -START
-SIXTEEN
&TYPE ...OPTGRAPH.... CLASSICAL ANALYSIS OF OPTSYS OUTPUT
SET LDRTELS 10
GLOBAL TXTLIB DISLIBVS 92DISLIB 92INTLIB VFORTLIB GRFLIB
NONIMSL CMSLIB
SET LDRTBLS 10
FILEDEF 01 DISK OPTGRNO DATA A1 (PERM
FILEDEF 02 DISK OPGRAPH LISTING A1
FILEDEF 03 PRINTER (RECFM FA BLKSIZE 133 PERM
FILEDEF 04 DISK OPTGRCM DATA A1 (PERM
FILEDEF 06 TERM (RECFM FA BLKSIZE 133
FILEDEF 05 TERM (RECFM FA BLKSIZE 80
FILEDEF 10 DISK OPTGRCL DATA A1 (PERM
FILEDEF 11 DISK LBLANK TMP A3 (RECFM F LRECL 2400 BLKSIZE

```

```

2400 XTENT 600
FILEDEF 12 DISK ISSCCMAP MAPDTA * (RECFM F LRECL 400
    BLKSIZE 400
FILEDEF 13 DISK TABLET TMP A3 (RECFM VS LRECL 208 BLKSIZE
    208 XTENT 1000
FILEDEF 14 DISK LBLANK MAPDTA &LBMODE (RECFM VS LRECL 608
    BLKSIZE 608
FILEDEF 17 DISK DISSPLA SYSUT1 A3 (RECFM FB LRECL 2000
    BLCK 2000 XTENT 500
FILEDEF 18 DISK DISSPLA METAFILE A4 (RECFM VBS LRECL 19065
    BLCK 19069

```

OPTGRAPH

-META

* * *

* CHECK FOR FILE "DISSPLA METAFILE A4" ON *

* THE USER'S DISK BEFORE GOING TO DISSPOP *

* * *

RENAME DISSPLA METAFILE A4 OPTSYS METAFILE A4

&IF &RETCODE NE 0 &GCIO -START

RENAME OPTSYS METAFILE A4 DISSPLA METAFILE A4

-EIGHT

CLRSCRN

&BEGTYPE -ENDTEN

DC YOU WANT A VRSTEC PLOTTER SMOOTH COPY OF THE
THE DISSPLA METAFILE THAT YOU JUST CREATED?

(Y OR N)

-ENDTEN

&READ VARS &ANS

&IF .&ANS EQ .Y &GOTO -NINTH

&IF .&ANS EQ .N &GOTO -START

&GOTO -EIGHT

-NINTH

EXEC DISSPOP VRSTEC

CLRSCRN

&BEGTYPE -ENDTWELVE

YOUR GRAPH(S) CAN BE PICKED UP AT THE COMPUTER CENTER.

THE GRAPH(S) WILL BE ADDRESSED TO "POP (USER ID)".

-ENDTWELVE

CP SLEEP 5 SEC

&GOTO -START

APPENDIX B
OPTSYSX PROGRAM LISTING

This portion of the thesis contains the OPTSYSX FORTRAN program (88 pages).

```

C-----
C OPTSYSX
C BY JCHN G. HODEN
C
C THIS PROGRAM IS A COMPLETELY INTERACTIVE
C OPTIMAL SYSTEMS CONTROL DESIGN/SYNTHESIS
C PROGRAM CAPABLE OF HANDLING VERY LARGE (80X80) +
C MULTIVARIABLE SYSTEMS OF LINEAR EQUATIONS.
C
C VERSION 1.08 11 MAR 1984
C MODIFIED BY AL DIEL
C MODIFIED BY M.H. LAPTAS
C
C IMPLICIT REAL*8(A-H,C-Z)
C
C INTEGER IANS,IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFCFW,IE,IDEBUG,ISET,
C 1IPSD,IYU,INCRM,NS,NC,NOB,NG,IREG,IDSTAB,IRET,NRCW,NCOL,ISAF,ISAH,
C 2SAG,IGAM,IRCMAT,ZERO,ONE,ITFX
C
C INCREASED DIMENSIONS -- LARGE ORDER SYSTEM.
C
C DIMENSION ACL(98,98),B(3,3),BA(98,98),CI(98),CR(98),CQ(58,98),
C *CWI(98),CWR(98),FBGC(3,98),FBGE(98,13),G(58,98),GM(58,98),
C *CPR(98,98),RC(13,13),SC(98,98),WR(98),WI(196),W1(98,98),
C *CWM(98,98),X(196,1),GN(98,98),HO(13,98),DI(196),D2(196),
C *CRM(196,196),Q(1,1),CAM(98,1),WNCRM(98,98),WNCRM1(98,98),
C *CDESTAB(98),AA(98,98),BM(98,3),CM(13,98),D(13,2),DSTCRE(58,98),
C *CJCF(196),RES(196),AY(98,98),BB(196),CC(196),CP(58),GW(196,1),
C *CGV(196,13),FY(13,196),HU(3,196),PRTT(16,16)
C
C STANDARD PROGRAM DIMENSIONS.
C
C DIMENSION ACL(32,32),B(3,3),FBGC(3,32),FBGE(32,32),G(32,32),
C 1I(32),CWR(32),SC(32,32),WR(64),WI(64),W1(32,32),W2(32,32),
C 2I(32,32),FO(32,32),D1(64),D2(64),RM(64,64),Q(32,32),CAM(32,32),
C 3GN(32,32),WNCRM1(32,32),DSTCRE(32,32),JCF(64),RES(64),AY(32,32),
C 4NORM(32,32),WNCRM(32,32),BB(196),CC(196),CP(58),GW(196,1),
C 5D(32,32),DSTCRE(32,32),JCF(64),RES(64),AY(32,32),BB(64),
C 6I(32),GW(64,64),GV(64,64),HU(64,64),PRTT(16,16)
C
C EQUIVALENCE (W1(1,1),GW(1,1)), (W1(1,1),GV(1,1)), (W2(1,1),
C 1,1), (W2(1,1),HU(1,1))
C
C COMMON /PRCG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEBUG,
C 1UG,ISET,IREG,IPSD,IYU,INCRM
C-----

```

```

C-----DATA IY/Y',IZ/'N'/-----OPT00490
C-----SUPPRESS INDIVIDUAL UNDERFLOW, OVERFLOW, DIVIDE CHECK, AND DECIMAL-----OPT00500
C-----CCNVERT ERROR MESSAGES; PROVIDE SUMMARY OF ERRORS ONLY.-----OPT00510
C-----OPT00520
C-----OPT00530
C-----CALL ERRSET (207,256,-1,1,1,209)-----OPT00540
C-----CALL ERRSET (215,256,-1,1,1)-----OPT00550
C-----INITIALIZE SAVE FLAGS.-----OPT00560
C-----ISAF=0-----OPT00570
C-----ISAG=0-----OPT00580
C-----ISAH=0-----OPT00590
C-----IGAM=0-----OPT00600
C-----ISAA=0-----OPT00610
C-----ISAB=0-----OPT00620
C-----ISET=0-----OPT00630
C-----OPT00640
C-----OPT00650
C-----OPT00660
C-----CALL FRTCMS ('CLRSCRN ')-----SCRNA-----OPT00670
C-----WRITE (5,88E)-----OPT00680
C-----CALL RDINT (IANS)-----OPT00690
C-----IF (IANS.EQ.1) GO TO 20-----OPT00700
C-----IF (IANS.EQ.2) GO TO 10-----OPT00710
C-----GO TO 5-----OPT00720
C-----OPT00730
C-----CALL FRTCMS ('CLRSCRN ')-----SCRN1-----OPT00740
C-----WRITE (5,88C)-----OPT00750
C-----CALL RDCHAR (IANS)-----OPT00760
C-----IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 30-----OPT00770
C-----GO TO 40-----OPT00780
C-----WRITE (5,88C)-----OPT00790
C-----GO TO 20-----OPT00800
C-----CCNTINUE-----OPT00810
C-----IF (IANS.EQ.IZ) GO TO 560-----OPT00820
C-----OPT00830
C-----CALL FRTCMS ('CLRSCRN ')-----SCRN2-----OPT00840
C-----WRITE (5,90C)-----OPT00850
C-----CALL RDCHAR (IANS)-----OPT00860
C-----IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60-----OPT00870
C-----GO TO 70-----OPT00880
C-----WRITE (5,88C)-----OPT00890
C-----GO TO 50-----OPT00900
C-----CCNTINUE-----OPT00910
C-----IF (IANS.EQ.IZ) GO TO 560-----OPT00920
C-----CALL FRTCMS ('CLRSCRN ')-----ISET-----OPT00930
C-----WRITE (5,91C)-----OPT00940
C-----OPT00950
C-----OPT00960

```

CALL RDCHAR (IANS)	OPT00970
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 90	OPT00980
GO TO 100	OPT00990
WRITE (5, 88C)	OPT01000
GO TO 80	OPT01010
C CONTINUE	OPT01020
IF (IANS.E6.IY) ISET=1	OPT01030
C-----	OPT01040
C INITIALIZE SYSTEM FLAGS.	= OPT01050
C-----	OPT01060
1C C CONTINUE	OPT01070
IRSET=0	OPT01080
IQL=0	OPT01090
IQ=0	OPT01100
IR=0	OPT01110
ISS=0	OPT01120
IM=0	OPT01130
ITF1=0	OPT01140
ITF2=0	OPT01150
ITF3=0	OPT01160
IFDFW=0	OPT01170
IF=0	OPT01180
IDSTAB=0	OPT01190
IDDEBUG=0	OPT01200
IPSD=0	OPT01210
IYU=0	OPT01220
INCRM=0	OPT01230
IREG=0	OPT01240
NS=0	OPT01250
NC=0	OPT01260
NOB=0	OPT01270
NG=0	OPT01280
IRDMAT=0	OPT01290
C-----	OPT01300
CALL FRICMS ('CLRSCRN ')	OPT01310
WRITE (5, 57C)	OPT01320
CALL RDINT (IANS)	OPT01330
ICL=IANS-1	OPT01340
IF (IOL.EQ.2) GO TO 350	OPT01350
IF (IOL.EQ.3) GO TO 200	OPT01360
C-----	OPT01370
CALL FRICMS ('CLRSCRN ')	OPT01380
WRITE (5, 58C)	OPT01390
CALL RDCHAR (IANS)	OPT01400
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 120	OPT01410
GO TO 130	OPT01420
WRITE (5, 88C)	OPT01430
GO TO 110	OPT01440

250	GO TO 260	OPT01930
	WRITE (5,88C)	OPT01940
260	GO TO 240	OPT01950
	CCNT INUE	OPT01960
	IF ((IANS.EQ.IY) IDSTAB=1	OPT01970
	IF ((IANS.EQ.IZ) IDSTAB=0	OPT01980
C	-----ICEBUG-----	OPT01990
270	WRITE (5,68C)	OPT02000
	CALL RDCCHAR (IANS)	OPT02010
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 280	OPT02020
280	GO TO 290	OPT02030
	WRITE (5,88C)	OPT02040
290	GO TO 270	OPT02050
	CCNT INUE	OPT02060
	IF ((IANS.EQ.IY) IDEBUG=1	OPT02070
	IF ((IANS.EQ.IZ) IDEBUG=0	OPT02080
300	CCNT INUE	OPT02090
C	-----IREG-----	OPT02100
320	CALL FRICMS ('CLRSCRN ')	OPT02110
	WRITE (5,71C)	OPT02120
	CALL RDCCHAR (IANS)	OPT02130
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 330	OPT02140
330	GO TO 340	OPT02150
	WRITE (5,88C)	OPT02160
340	GO TO 320	OPT02170
	CCNT INUE	OPT02180
	IF ((IANS.EQ.IY) IREG=1	OPT02190
	IF ((IANS.EQ.IZ) IREG=0	OPT02200
350	CALL RDMATF(NS,NC,NCB,NG,ISAF,ISAG,ISAH,IGAM,ISAB,IRDMAT)	OPT02210
	IF ((ISAF.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 352	OPT02220
C	-----NS-----	OPT02230
	CALL FRICMS ('CLRSCRN ')	OPT02240
	WRITE (5,72C)	OPT02250
	CALL RDCREAL (ANSR)	OPT02260
	NS=IDINT (ANSR)	OPT02270
352	IF ((IOL.EQ.2) GO TO 360	OPT02280
	IF ((ISAG.EQ.1).AND.(IPDMAT.EQ.1)) GO TO 354	OPT02290
C	-----NC-----	OPT02300
	WRITE (5,73C)	OPT02310
	CALL RDCREAL (ANSR)	OPT02320
	NC=IDINT (ANSR)	OPT02330
354	IF ((ISAH.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 356	OPT02340
C	-----NCB-----	OPT02350
	WRITE (5,74C)	OPT02360
	CALL RDCREAL (ANSR)	OPT02370
	NOB=IDINT (ANSR)	OPT02380
356	IF ((IGAM.EQ.1).AND.(IRDMAT.EQ.1)) GO TO 360	OPT02390
C	-----NG-----	OPT02400


```

C-----OPT03370
358 IF(IRET.EQ.1) GO TO 370
CALL WRMTAT(BA,G,HO,GAM,FBGC,FBGE,AY,B,NS,NC,NOB,NG)
C-----OPT03380
370 WRITE (5,82C)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 380
GO TO 390
380 WRITE (5,88C)
GO TO 370
390 CONTINUE
C-----RESET OPGRAPH DATA FILE
IF(IANS.EQ.IZ) GO TO 395
END FILE 10
REWIND 10
END FILE 1
REWIND 1
END FILE 4
REWIND 4
C-----OPT03490
395 IF (IANS.EQ.IY) GO TC 400
IF (IANS.EQ.IZ) GO TC 560
C-----ISAF-----OPT03500
400 CONTINUE
IF (IRET.EQ.1) GO TC 10
IF (ISET.EQ.1) GO TC 10
CALL FRTCMS ('CLRSCRN ')
410 WRITE (5,840)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 420
GO TO 430
420 WRITE (5,88C)
GO TO 410
430 CONTINUE
IF (IANS.EQ.IY) ISAF=1
IF (IANS.EQ.IZ) ISAF=0
C-----ISAH-----OPT03610
440 IF (NOB.EQ.0) GO TO 470
CALL FRTCMS ('CLRSCRN ')
WRITE (5,85C)
CALL RDCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 450
GO TO 460
450 WRITE (5,88C)
GO TO 440
460 CONTINUE
IF (IANS.EQ.IY) ISAH=1
IF (IANS.EQ.IZ) ISAH=0
OPT03620
OPT03630
OPT03640
OPT03650
OPT03660
OPT03670
OPT03680
OPT03690
OPT03700
OPT03710
OPT03720
OPT03730
OPT03740
OPT03750
OPT03760
OPT03770
OPT03780
OPT03790
OPT03800
OPT03810
OPT03820
OPT03830
OPT03840

```


470	CCNT INUE		ISAG	OPT03850
C	IF (NC.EQ.C) GO TO 510			OPT03860
	CALL FRTCMS ('CLRSCRN')			OPT03870
480	WRITE (5,88C)			OPT03880
	CALL RDCHAR (IANS)			OPT03890
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 490			OPT03900
	GO TO 500			OPT03910
490	WRITE (5,88C)			OPT03920
	GO TO 480			OPT03930
500	CCNT INUE			OPT03940
	IF (IANS.EC.IY) ISAG=1			OPT03950
	IF (IANS.EC.IZ) ISAG=0			OPT03960
510	CCNT INUE		IGAM	OPT03970
C	IF (NG.EQ.C) GO TO 550			OPT03980
	CALL FRTCMS ('CLRSCRN')			OPT04000
520	WRITE (5,87C)			OPT04010
	CALL RDCHAR (IANS)			OPT04020
	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 530			OPT04030
	GO TO 540			OPT04040
530	WRITE (5,88C)			OPT04050
	GO TO 520			OPT04060
540	CCNT INUE			OPT04070
	IF (IANS.EC.IY) IGAM=1			OPT04080
	IF (IANS.EC.IZ) IGAM=0			OPT04090
550	CCNT INUE		ISAA	OPT04100
C	CALL FRTCMS ('CLRSCRN')			OPT04110
551	WRITE (5,872)			OPT04120
	CALL RDCHAR (IANS)			OPT04130
	IF ((IANS.EC.IY).OR.(IANS.EC.IZ)) GO TO 553			OPT04140
	WRITE (5,88C)			OPT04150
	GO TO 551			OPT04160
553	CCNT INUE			OPT04170
	IF (IANS.EC.IY) ISAA=1			OPT04180
	IF (IANS.EC.IZ) ISAA=0			OPT04190
C	CALL FRTCMS ('CLRSCRN')		ISAB	OPT04200
555	WRITE (5,874)			OPT04210
	CALL RDCHAR (IANS)			OPT04220
	IF ((IANS.EC.IY).OR.(IANS.EC.IZ)) GO TO 557			OPT04230
	WRITE (5,88C)			OPT04240
	GO TO 555			OPT04250
557	CCNT INUE			OPT04260
	IF (IANS.EC.IY) ISAB=1			OPT04270
	IF (IANS.EC.IZ) ISAB=0			OPT04280
	GO TO 10			OPT04290
				OPT04300
				OPT04310
				OPT04320

```

C-----WRITE (5,92C)-----TERMINATE-----OPT04330
560      STOPOPT04340
C-----OPT04350
885      FORMAT (5X,22HIN CASE OF PROBLEMS CALL EXT 2826,///,10X,42H1 DEFINOPT04360
      1NES MATRIX EQUATIONS USED IN OPTSYS,/,15X,22HAND ALTERNATE DATA INOPT04370
      2PUT METHOD,/,10X,26H2 START INTERACTIVE INPUT,/,15X,13HENTER OPT04380
      3 1 OR 2,/,5X,59H NORMALLY TWO SUCCESSIVE NULL ENTERS TERMINATES TOPT04390
      4HE PROGRAM.)OPT04400
890      FORMAT (5X,59HOPTSYSX IS A COMPLETELY INTERACTIVE OPTIMAL SYSTEMSOPT04410
      1CONTROL,/,8X,55HPROGRAM. IT WILL SOLVE NUMEROUS CONTROL PROBLEMS OPT04420
      2N THE,/,8X,45HFEEDBACK TYPES OF SYSTEMS CONTROL PROBLEMS OPT04430
      3,35HXDOT = H3*X + D3*U + GAMMA*(W+W0),/,20X,22HMEASUREMENT EQUAOPT04440
      4TION,/,15X,21HZ = 1/2 * INTEGRAL (Y9, AG*Y + U9* B8*U)DT,OPT04450
      5ANCE INDEX,/,15X,42HJ = 1/2 * INTEGRAL (Y9, AG*Y + U9* B8*U)DT,OPT04460
      6,/,20X,32HSTATE WISH TO CONTRY,/,5X,49HALTHOUGH OPTSYSX IS SPECIFIOPT04470
      7,FORMAT (25X,14H--DATA ENTRY--,,5X,48HALL MATRIX DATA INTERACTIVELY, SEVEOPT04480
      1CALLY DESIGNED TO READ,/,5X,48HALL MATRIX DATA INTERACTIVELY, SEVEOPT04490
      2RAL ALTERNATE,/,5X,31HMETHODS ARE AVAILABLE TOC USERS:,,10X,43HMEOPT04500
      3THOD 1--THE DATA FILES,/,10X,50HMETRICALLY DEFINE WITHIN SUBROUTINE "GAMMAOPT04510
      4N SEPARATES MAY BE,/,13X,45HEXPLICITLY DEFINE WITHIN SUBROUTINE "GAMMAOPT04520
      5" MAT,/,10X,52HNO THE: IN EITHER CASE, THE USER SHOULD OBTAIN A COPT04530
      6ETUP,/,17X,24HCF THE PROGRAM LISTING AND EXAMINE,/,17X,35HTHE EXAMPOPT04540
      7OPY,/,17X,24HCF THE PROGRAM LISTING AND EXAMINE,/,17X,35HTHE EXAMPOPT04550
      8LTYPE,CONTAINS "YES" OR "NO",/,"SETUP",.8,/,10X,45HDC YOU WISH TO CONTINUE?OPT04560
      9FORMAT (/,5X,46HDC YOU WISH TO INPUT THE "F", "G", AND "GAMMA",/,OPT04570
      110X,40HMATRIX FROM SUBROUTINE "SETUP" IAW THE,/,15X,19HTYPE "YES" OR "NO",/,"GAMMA",/,OPT04580
      2SCRIBED ON THE PREVIOUS SCREEN,/,15X,19HTYPE "YES" OR "NO",/,"GAMMA",/,OPT04590
      3FORMAT (25X,24HGENERATION,/,22X,35HOPTSYSX OPEN-LOOP EIGENSYSTEM CALCULATION 1--SYSTOPT04600
      1EM ANALYSIS WITH 2--SYSTEM ANALYSIS WITH 3--CPEN-LOOP EIGENSYSTEM FOPT04610
      2,/,10X,42HOPTIONS,/,10X,39HOPTIONS,/,22X,25HFEIGENSYSTEM FOPT04620
      3SYSTEM CALCULATION,/,10X,39HOPTIONS,/,22X,25HFEIGENSYSTEM FOPT04630
      4OUND,/,22X,23HAND PRGMM TERMINATES,/,22X,35H--MCCAL DISTRIBUTION MATRFOPT04640
      5COWS IMMEDIATELY,/,10X,48HOPTION 4--REGULATOR SYNTHESIS,/,22X,35H--MCCAL DISTRIBUTION MATRFOPT04650
      6COWS IMMEDIATELY,/,10X,48HOPTION 4--REGULATOR SYNTHESIS,/,22X,35H--MCCAL DISTRIBUTION MATRFOPT04660
      7,225HOR STEADY-STATE ANALYSIS,/,15X,30HSELECT AN OPTION: 1,2,3, OPT04670
      8R 4.)OPT04680
      9FORMAT (/,5X,46HDC YOU DESIRE RMS VALUES OF STATE AND CONTRCL,/,OPT04690
      1,10X,19HTYPE "YES" OR "NO",/,"GAMMA",/,OPT04700
      2FORMAT (/,20X,39HOPTSYSX LQR/CLASSICAL OPTICS:,,10X,43HCPCTICN 1OPT04710
      1--OPTIMAL CR "K",/,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04720
      2MAL FILTER CR "K",/,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04730
      3MAL FILTER CR "K",/,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04740
      4,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04750
      5REGULATOR,/,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04760
      6U,/,10X,43HOPTION 2--OPT04770
      7,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04780
      8,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04790
      9,22X,13HMATRIX INPUT,/,10X,43HOPTION 2--OPT04800

```



```

      WITH EXTERNAL "C" AND "K" MATRIX INPUT.,//,10X
      32HSELECT AN OPTION: 1, 2, 3, OR 4.)
      FORMAT (//,5X,50HDC YOU WISH TO DETERMINE THE STEADY-STATE RESPONSE
      1E,/,8X,27HFCR A CONSTANT DISTURBANCE?,//,10X,19HTYPE "YES" CR "NO"
      2.)
      FORMAT (5X,47HDO YOU WISH TO DETERMINE THE MICAL DISTRIBUTION,/,8X
      1,18HAND GA IN MATRICES?,//,10X,19HTYPE "YES" CR "NO".)
      FORMAT (//,5X,36HOPEN-LOOP TRANSFER FUNCTION COMPUTED.,//,10X,48H
      1OPTION 1 -- NC OPEN-LOOP TRANSFER FUNCTION COMPUTED.,//,10X,42H
      2CN 2 -- POLES AND RESIDUES COMPUTED.,//,10X,45HOPTION 3 --
      3ONLY RESIDUES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3,
      4AND RESIDUES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3,
      5//,10X,32HIF ANALYSIS CAUSES MODAL OUTPUT,//,5X,61HPCLES AND
      6FORMAT (//,5X,32HNOISE TRANSFER FUNCTION COMPUTED.,//,10X,49H
      1 -- NO NOISE TRANSFER FUNCTION COMPUTED.,//,10X,48HOPTION 2 --
      2POLES AND RESIDUES COMPUTED.,//,10X,42HOPTION 3 -- ONLY
      3PLES AND RESIDUES COMPUTED.,//,10X,45HOPTION 4 -- ONLY
      4LES COMPUTED.,//,10X,32HSELECT AN OPTION: 1, 2, 3, OR 4.)
      1OPTION 1 -- NO COMPLETS, AND ZEROS COMPUTED.,//,10X,49H
      2 -- POLES AND ZEROS COMPUTED.,//,10X,48HOPTION 2 --
      3ONLY RESIDUES COMPUTED.,//,10X,45HOPTION 3 -- ONLY
      4RESIDUES COMPUTED.,//,10X,45HOPTION 4 -- ONLY PCLES AND
      5ON CAN BE,/,22X,33HCCMPUTED ONLY IF BOTH A REGULATOR,/,22X,26H
      6FILTER ARE,/,22X,14HAND/CR INPUT.6,/,10X,32HSELECT AN
      7FORMAT (//,5X,39HWILL A FEED-FORWARD DISTRIBUTION MATRIX,/,5X,25H
      1"0" -- YES,15X,19HTYPE "YES" OR "NO".)
      1WHEN A MARKOV,63H THIS OPTICN DETERMINES THE CRITERIA FOR DECIDING
      2ATES THE ORDER,/,8X,54HCF "N" ZEROS OF THIS POLYNOMIAL OF EACH
      3R FUNCTION,/,8X,52HTEST TELLS HOW MANY EXTRA ROOTS EXIST AT Z=
      4OUT /,8X,47HTHIS IS CONSIDERED ZERO.,//,8X,47HTHE OPTICN
      50,DEFAULT,/,8X,47HTHIS IS PARAMEETER DESIRE A DIFFERENT MARKOV
      6,ESP=1,OE=6,/,10X,66HIF YOU DESIRE THE DEFAULT VALUE,/,10X,48H
      7,TYPE "0" ZEROS.)
      8,TYPE "0" ZEROS.)
      9FORMAT (//,5X,61HDC YOU DESIRE TO SYNTHESIZE A STABLE FILTER OR
      1REGULATOR,/,8X,34HDESTABILIZING THE ORIGINAL SYSTEM?,//,12X,52H
      2NOT SAME WORKS FOR FILTER OR REGULATOR,/,20X,17HIN T
      3HE SAME WORKS,/,8X,19HTYPE "YES" OR "NO".)
      1HOPMAT (5X,47HDO YOU WISH TO DETERMINE THE MICAL DISTRIBUTION,/,8X
      2X,19HTYPE "YES" OR "NO".)
      1OPTION 1 -- SPECTRAL DENSITY PSD& OPTION 1 ://,10X,53H
      1OPTION 1 -- SPECTRAL DENSITY PSD& OPTION 1 ://,10X,53H

```



```

CALL FRTCMS ('FILEDEF','03      ','DISK      ','X29A82      ',
1,CATA      ',A
-----
THIS IS AN EXAMPLE OF A 83 X 84 DATA FILE X29A83 DATA A1 READ FROM
A USER'S DISK AND CONVERTED (FROM A "DUMMY" ARRAY NAMED "DUM") TO A
SYMMETRIC ARRAY. THE FCRMAT STATEMENT MUST MATCH YOUR DISK DATA
FCRMAT OR THE PROGRAM WILL FAIL. NOTE: ALL PROGRAM DIMENSIONS
MUST BE ENLARGED ACCORDINGLY FOR A SYSTEM OF THIS SIZE.
-----
      READ (3,10C)
      READ (3,10C)
      DO 20 I=1,NS
      READ (3,50) (BA(I,J),J=1,NS)
      CCNTINUE
20-----
THESE ARE EXAMPLES OF SEVERAL POSSIBLE METHODS OF ARRAY GENERATION
WITHIN SUBROUTINE SETUP. THE "GAM" ARRAY WAS SET TO ZERO SINCE NO
"NOISE" WAS PRESENT, AND THE NON-ZERO ELEMENTS OF THE "G" ARRAY WERE
EXPLICITLY DEFINED. THEY COULD ALSO BE READ FROM FILES AS ABOVE.
-----
      DO 40 I=1,NS
      DO 30 J=1,NC
      GAM(I,J)=0.0D+00
      G(I,J)=0.0D+00
      CCNTINUE
      G(82,1)= 0.1000D+00
      G(52,1)= 0.362D+07
      G(77,1)=-0.1591D+02
      G(78,1)= 0.2448D+00
      G(79,1)= 0.2448D+00
      G(81,1)= 0.10C0D+00
-----
      READ (3,10C)
      DO 70 I=1,NS
      READ (3,50) (G(I,J),J=1,NC)
      CCNTINUE
70-----
THIS IS AN EXAMPLE OF ONE POSSIBLE METHOD OF ARRAY GENERATION
WITHIN THE PRGCGRAM ITSELF. FOR VERY LARGE DATA ARRAYS, THIS METHOD
MAY BE PREFERABLE TO SOME USERS OVER INTERACTIVE ENTRY OF EACH
INDIVIDUAL ELEMENT.
-----
      DO 2 I=1,11
      DO 1 J=1,82
      HO(1,J)
      HO(1,1)
      HO(2,75)
      = 0.0D+00
      = 0.11520D+00
      = 0.5730D+02
-----

```



```

C-----COMPENSATOR TF-----OPT07210
2C  IF (ITF3 .EQ. 0) GO TO 30 OPT07220
   IF (IREG .EQ. 0 .AND. (NC .NE. 0) GO TC 30 OPT07230
   WRITE (5,11C) OPT07240
   IRET=1 OPT07250
   RETURN OPT07260
   CONTINUE OPT07270
C-----NCISE TF-----OPT07280
3C  IF (ITF2 .EQ. 0) GO TO 40 OPT07290
   IF (NG.NE. 0 .AND. NC .NE. 0) GO TO 40 OPT07300
   WRITE (5,12C) OPT07310
   IRET=1 OPT07320
   RETURN OPT07330
C-----DESTABILIZATION RESTRICTIONS-----OPT07340
4C  IF (IDSTAB .EQ. 0) GO TO 50 OPT07350
   IF (NC .EQ. 0) GO TC 50 OPT07360
   IF (NG .NE. 0) IREG=1 OPT07370
   WRITE (5,13C) OPT07380
   IF (IREG .EQ. 1) GO TO 50 OPT07390
   IRET=1 OPT07400
   RETURN OPT07410
   CONTINUE OPT07420
C-----PSD INPUT-----OPT07430
5C  IF (IPSD .EQ. 0) GO TO 80 OPT07440
   IF (IPSD .LT. 0 .OR. IPSD .GT. 3) GO TO 60 OPT07450
   IF (IYU .LT. 0 .OR. IYU .GT. 3) GO TO 60 OPT07460
   IF (INORM .LT. 0 .OR. INORM .GT. NG+ND) GO TC 60 OPT07470
   GO TO 70 OPT07480
   WRITE (5,14C) OPT07490
   IRET=1 OPT07500
   RETURN OPT07510
7C  IF (IREG .EQ. 0 .AND. NC .NE. 0) GO TO 80 OPT07520
   WRITE (5,15C) OPT07530
   IRET=1 OPT07540
   RETURN OPT07550
   CONTINUE OPT07560
   RETURN OPT07570
C-----FORMAT (//,EX,49H H - MATRIX MUST BE INPUT, I.E. "NC" MUST BE > 0. OPT07580
90  FORMAT (//,EX,49H H - MATRIX MUST BE INPUT, I.E. "NC" MUST BE > 0. OPT07590
   1,///) OPT07600
100  FORMAT (//,EX,46H(G) MATRIX MUST BE INPUT, I.E. NC MUST BE > 0.,/, OPT07610
   110X,26HTO CCMPUTE OPEN LOOP TO F.,//) OPT07620
110  FORMAT (//,EX,48HREGULATOR AND FILTER SYNTHESIS MUST BE REQUESTED, OPT07630
   1/,5X,44HIN THE SAME RUN TO COMPUTE COMPENSATOR T. F.,/,5X,47H I.E. OPT07640
   2IPEG.MUST = 0.: "NC" AND "NG" MUST BE > 0.,//) OPT07650
120  FORMAT (//,EX,51HNCISE T. F. CALCULATED ONLY WHEN REGULATOR DESIGN OPT07660
   1ED.,/,5X,47H I.E. IREG.MUST = 1.: "NC" AND "NG" MUST BE > 0.,//) OPT07670
130  FORMAT (//,EX,47HDESTABILIZATION OPTION DESIGNED FOR A REGULATOR, OPT07680
   1,///)

```



```

1,5X,38HOR FILTER BUT NOT BOTH SIMULTANEOUSLY.,//,5X,55HIF "NC" > 00PT07690
2. THE REGULATOR OPTION IS AUTOMATICALLY SET ,//) OPT07700
FORMAT (//,5X,49H ***** INCONSISTENT PSD INPUT FLAGS ***** OPT07710
1,//) OPT07720
150 FORMAT (//,5X,44H BOTH A REGULATOR AND FILTER MUST BE RESIDENT,/,10X,42HI.E. IREG OPT07730
1X,42HTO COMPUTE THE PSD OF A CONTROLLED SYSTEM ,/,10X,42HI.E. IREG OPT07740
2 MUST BE 0. AND "NC" MUST BE > 0.,//) OPT07750
END OPT07760
C===== OPT07770
SUBROUTINE INNER (NS,NC,NO,NG,N2,ACL,B,BA,CI,CR,CQ,CWI,CWR,D,FBGC, OPT07780
1FBGE,G,GAM,GM,GN,HC,D1,D2,PRO,RC,Q,SC,WR,WI,W21,X,WNORM,WNO OPT07790
2RMI,DESTAB,AA,BM,CM,JCF,RES,AY,BB,CC,CP,GW,GV,HY,HU,DS+CRE,ISAF,IS OPT07800
3AH,ISAG,IGAM,IRET,PRTT,NROW,NCOL,IRDMAT,ISAA,ISAB) OPT07810
C===== OPT07820
IMPLICIT REAL*8(A-H,C-Z) OPT07830
C----- OPT07840
DIMENSION ACL(NS,NS),B(NC,NC),BA(NS,NS),CI(NS),CR(NS),CQ(NS,NS),CW OPT07850
1I(NS),CWR(NS),FBGC(NC,NS),FBGE(NS,NO),G(NS,NS),GM(NS,NS),PRC(NS,NS),NS OPT07860
2),RC(NC,NO),SC(NS,NS),WR(N2),WI(N2),W21(NS,NS),X(N2,N2),X(N2,N2),X(N2,N2) OPT07870
3,GN(NS,NS),FO(NO,NS),DI(N2),D2(N2),RM(N2,N2),C(NG,NG),D(NC,NC),GAM OPT07880
4(NS,NG),WNORM(NS,NS),WNORMI(NS,NS),DESTAB(NS),AA(NS,NS),BM(NS,NC),CP(NS,NC), OPT07890
5CM(NO,NS),JCF(N2),RES(N2),AY(NO,NO),BB(N2),CC(N2),CP(NS),GW(N2,NG), OPT07900
6,GV(N2,NG),FY(NO,N2),HU(NC,N2),DSTORE(NS,NS),PRTT(16,16) OPT07910
C----- OPT07920
COMMON /PRCG/ IOL,IQ,IR,ISS,IM,ITF1,ITF2,ITF3,IFDFW,IE,IDSTAB,IDEB OPT07930
1UG,ISET,IREG,IPSD,IYL,INORM OPT07940
C----- OPT07950
REAL#4 FMT(20) OPT07960
C----- OPT07970
OUTPUT OPTIONS----- OPT07980
IOL=1 IF THE OPEN LOOP EIGENSYSTEM IS DESIRED--CITHEWISE IOL=0 OPT07990
IQ=1 IF THE RMS VALUES OF THE CCNTROL AND STATE ARE TO BE FOUND OPT08000
IR=0 IF OPTIMAL FILTER AND STATE ARE TO BE FOUND OPT08010
IR=1 IF EXTERNAL C MATRIX IS SUPPLIED OPT08020
IR=2 IF EXTERNAL K IS SUPPLIED OPT08030
IR=3 IF EXTERNAL C AND K ARE TO BE DETERMINED OPT08040
ISS=1 IF STEADY STATE VALUES ARE TO BE DETERMINED OPT08050
IM=1 IF MODAL STATES DESIRED OPT08060
C----- OPT08070
NSQ=NS*NS OPT08080
MH=NS OPT08090
M=N2 OPT08100
CALL CHECK (EPS,NC,NG,NO,IRET) OPT08110
IF (IRET.EQ.1) RETURN OPT08120
IF (ISET.EQ.1) GO TO 20 OPT08130
CALL RCMAT (PA,G,HQ,GAM,FBGC,FBGE,AY,B,NS,NC,NG,IRDMAT) OPT08140
CALL READF (NS,ISAF,EA) OPT08150
IF (IDSTAB.EQ.0) GC TO 10 OPT08160
WRITE (5,18C0)

```

```

CALL RCREAL (ANSR)
CSTAB=ANSR
DO 10 I=1,NS
  CESTAB(I)=CSTAB
CCNT INUE
GC TO 30
CALL SETUP (BA,G,GAM,HO,NS,NC,NG,NO)
CONTINUE (6,1330)
DO 40 I=1,NS
  WRITE (6,1330) (BA(I,J),J=1,NS)
  IF (IDSTAB.EQ.0) GO TO 50
  WRITE (6,1400)
  WRITE (6,1330) (DESTAB(I),I=1,NS)
CONTINUE
-----EIGENSYSTEM OF THE OPEN LOOP DYNAMICS-----
IF (IOL.EQ.C.AND.IQ.EQ.0) GO TO 90
IF (IOL.EQ.C.AND.NC.NE.0) GO TO 90
DO 60 I=1,NS
  DO 60 J=1,NS
    GN(I,J)=BA(I,J)
  CALL BALANC (NS,NS,GN,LOW,HIGH,D1)
  CALL ORTHES (NS,NS,LCW,HIGH,GN,D2)
  CALL ORTRAN (NS,NS,LCW,HIGH,GN,CWR,CWI,SC,IERR)
  CALL HQR2 (NS,NS,LOW,HIGH,GN,CWR,CWI,SC,IERR)
  IF (IERR.NE.0) CALL EEXIT (NS,GN,IERR)
  CALL BALBAK (NS,NS,LCW,HIGH,D1,NS,SC)
  CALL NORMALIZE AND PRINT OPEN LOOP EIGENSYSTEM
  IWRITE=1
  CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNCRM,WNCRMI,PC,CM,
1NO,NS)
  DO 64 I = 1,NS
    WRITE (1,2000) CWR(I), CWI(I)
    WRITE (1,2000) CWR(I), CWI(I)
  CCNT INUE
  GC TO 70
  IF (IOL.EQ.2) RETURN
  IF (IQ.EQ.0.OR.(NC.NE.0.OR.IDSTAB.GT.0)) GO TO 50
  DO 70 I=1,NS
    IF (CWR(I).LT.0.) GC TO 70
  WRITE (5,1450)
RETURN
CCNT INUE
IF (IOL.EQ.2) GO TO 130
DO 80 I=1,NS
  DO 80 J=1,NS
    W1(I,J)=SC(I,J)

```

```

          CALL MINV (NSQ,W11,NS,DDD,D1,D2)
          CCNT INUE
          IF (IDSTAB.EQ.0) GO TO 130
          FORM U * DIAG(DESTAB) * U-INV-----
C-----
100      DO 100 J=1,NS
          DO 100 I=1,NS
          AA(I,J)=WNCFM(I,J)*DESTAB(J)
          DO 120 I=1,NS
          DO 120 J=1,NS
          CDD=0.00
          DO 110 K=1,NS
          CDD=DDC+AA(I,K)*WNORMI(K,J)
          DSTORE(I,J)=DDD
          BA(I,J)=BA(I,J)+DDD
          CCNT INUE
          IF (NO.EQ.0) GO TO 145
          IF (ISET.EC.1) GO TO 135
          CCALL INUE (NO,NS,ISAH,HO)
          WRITE (6,1440)
          DO 140 I=1,NC
          WRITE (6,1350) (HO(I,J),J=1,NS)
          IF (IM.NE.1) GO TO 150
          CCALL INUE (WNORM,HO,CM,NS,NO,NS,2)
          CCNT INUE
          IF (IFCFW.EC.0) GO TO 170
          CALL READD (NO,NC,D)
          WRITE (6,1470)
          DO 160 I=1,NC
          WRITE (6,1350) (D(I,J),J=1,NC)
          CONT INUE
          NOB=0
          IF (NC.EQ.0) GO TO 550
          IF (IOL.EQ.2) GO TO 270
          IF (IR.NE.1.AND.IR.NE.3) GO TO 210
          IF (ISET.EC.1) GO TO 180
          CALL INUE (NS,NC,ISAG,G)
          CCNT INUE
          CALL READDFE (NC,NS,FEGC)
          WRITE (6,1400)
          DO 190 I=1,NS
          WRITE (6,1350) (G(I,J),J=1,NC)
          IF (IM.NE.1) GO TO 200
          CALL INUE (WNORMI,G,BM,NS,NS,NC,0)
          CONT INUE
          GO TO 330
          DO 220 I=1,NS
          DO 220 J=1,NS

```

```

OPT08650
OPT08660
OPT08670
OPT08680
OPT08690
OPT08700
OPT08710
OPT08720
OPT08730
OPT08740
OPT08750
OPT08760
OPT08770
OPT08780
OPT08790
OPT08800
OPT08810
OPT08820
OPT08830
OPT08840
OPT08850
OPT08860
OPT08870
OPT08880
OPT08890
OPT08900
OPT08910
OPT08920
OPT08930
OPT08940
OPT08950
OPT08960
OPT08970
OPT08980
OPT08990
OPT09000
OPT09010
OPT09020
OPT09030
OPT09040
OPT09050
OPT09060
OPT09070
OPT09080
OPT09090
OPT09100
OPT09110
OPT09120

```


92

```

450      WRITE (6,1570)
      CALL RAPRNT (NS,NS,NS,9,SC,4,'(9(1X,1PD13.6))')
      CCNT INUE
C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
      IF (IDSTAB.EQ.0) GC TO 470
      DO 460 I=1,NS
      BA(I,I)=BA(I,I)-DESTAB(I)
      IR=1
470      CCNT INUE
C-----CALCULATION OF FEEDBACK GAIN-----
C-----FEEDBACK GAINS--> U = -(BINVERSE)*GT*GN&-----
C-----CALCULATE GT-----
      DO 490 I=1,NC
      DO 490 J=1,NS
      PRO(I,J)=0.0
      DO 480 K=1,MH
      PRO(I,J)=PRO(I,J)+G(K,I)*GN(K,J)
480      FBGC(I,J)=-PRO(I,J)/B(I,I)
490      IF (IDSTAB.EQ.1) GC TO 500
C-----NORMALIZE AND PRINT OPT. REG. CLOSED LOOP EIGENSYSTEM-----
      IWRITE=2
      CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNCRMI,FBGC,
      1AA,NC,NS)
C-----THE OPTIMUM FEEDBACK CONTROL GAINS-----
500      WRITE (6,1580)
      DO 510 I=1,NC
510      WRITE (6,1590) (FBGC(I,J),J=1,NS)
C-----COMPUTE MCCAL C MATRIX OPEN LOOP U-INVERSE SAVED IN WNORM&-----
      IF (IM.NE.1) GO TC 530
C-----IN COMPUTING MCIAC RECCOMPUTE U OPEN LOOP SINCE WNORM USED TO STORE
C U & U-INV FOR CLOSED LOOP SYSTEMS; WNCRMI USED TO SAVE U-INV OPEN LOOP
C-----
      DO 520 I=1,NS
      DO 520 J=1,NS
      WNCRM(I,J)=WNORM(I,J)
520      CALL MINV (NSQ,WNORM,NS,DDD,D1,D2)
530      CALL MCDE (WNORM,FBGC,AA,NS,NC,NS,3)
      CCNT INUE
C-----THE CLOSED LOOP DYNAMICS MATRIX-----
      DO 550 I=1,NS
      DO 550 J=1,NS
      SUM=0.0
      DO 540 K=1,NC
      SUM=SUM+G(I,K)*FBGC(K,J)
540      ACL(I,J)=BA(I,J)+SUM
550      WRITE (6,1600)
      CALL RAPRNT (MH,MH,MH,5,ACL,4,'(5(1X,1PD13.6))')

```



```

560 IF (IR.NE.1.AND. IR.NE.3) GO TO 590
DO 560 I=1,NS
DO 560 J=1,NS
GN(I,J)=AC1(I,J)
CALL BALANC (NS,NS,GN,LOW,IHIGH,D1)
CALL ORTRAN (NS,NS,LCW,IHIGH,GN,D2)
CALL ORTR2 (NS,NS,LOW,IHIGH,GN,CWR,CWI,SC,IERR)
IF (IERR.NE.0) CALL EREXIT (NS,GN,SC,IERR)
CALL BALBAK (NS,NS,LCW,IHIGH,D1,NS,SC)
C-----NORMALIZE AND PRINT CLOSED LOOP SUBOPT. REG. EIGENSYSTEM-----
IWRITE=3
IWRITE=3
CALL CNORM (CWR,CWI,SC,NS,IWRITE,NSQ,DDD,D1,D2,WNORM,WNCRMI,FBGC,
1AA,NC,CNS)
DO 570 I=1,NS
IF (CWR(I).LT.0.0) GC TO 570
WRITE (5,1610)
RETURN
570 CCNT INUE
IF (IQ.NE.1) GO TO 590
DO 580 I=1,NS
DO 580 J=1,NS
W1(I,J)=SC(I,J)
CALL MINV (NSQ,W1,NS,DDD,D1,D2)
NCB=NO
IF (ING.EQ.0) RETURN
IF (ISET.EQ.1) GO TO 610
CALL ISEADG2 (NS,NG,IGAM,GAM)
CCNT INUE
IF (IOL.EQ.3) GO TO 620
CALL IREADQ (NG,Q)
DO 630 I=1,NS
WRITE (6,1420) (GAM(I,J),J=1,NG)
IF (IMCDE.NE.1) GO TO 640
CALL IMCDE (WNORMI,GAM,AA,NS,NS,NG,1)
CCNT INUE
IF (IOL.EQ.3) RETURN
WRITE (6,1420)
DO 650 I=1,NG
WRITE (6,1420) (Q(I,J),J=1,NG)
IF ((IQ.EQ.0).AND.(NG.EQ.0)) GO TO 1260
DO 660 I=1,NS
DO 660 J=1,NS
PRC(I,J)=O.0
DO 660 K=1,NG
PRC(I,J)=PRC(I,J)+Q(I,K)*GAM(J,K)
DO 570

```

```

OPT110570
OPT110580
OPT110590
OPT110600
OPT110610
OPT110620
OPT110630
OPT110640
OPT110650
OPT110660
OPT110670
OPT110680
OPT110690
OPT110700
OPT110710
OPT110720
OPT110730
OPT110740
OPT110750
OPT110760
OPT110770
OPT110780
OPT110790
OPT110800
OPT110810
OPT110820
OPT110830
OPT110840
OPT110850
OPT110860
OPT110870
OPT110880
OPT110890
OPT110900
OPT110910
OPT110920
OPT110930
OPT110940
OPT110950
OPT110960
OPT110970
OPT110980
OPT110990
OPT111000
OPT111010
OPT111020
OPT111030
OPT111040

```



```

74C RM(I+MH,J+MH)=-BA(J,I)
    RM(I,J+NS)=CC(I,J)
    GO TO 400
C-GO BACK TO 450 TO SET UP THE FILTER HAMILTONIAN: CALC.THE FILTER GAINS
750 CALL RGAIN (M,NS,NC,NCB,WR,WI,X,GN,GM ,RM,W21,D1,CR,CI,PRO,M+S,D2)
C-----CHECK EIGENVECTORS-----
    IF (IDEBUG.EQ. 0) GC TO 760
    WRITE (6,I=70)
    CALL RAPRNT (NS,NS,NS,9,PRO,4,'(9(1X,1PD13.6))')
760 CCNT INUE
    IF (IDSTAB.EQ. 1) GC TO 770
C-----NORMALIZE AND PRINT OPT. ESTIMATOR EIGENSYSTEM-----
    IWRITE=4
    CALL CNORM (CR,CI,PRC,NS,IWRITE,NSQ,DDD,D1,D2,WNCRM,WNORMI,HC,AA,
77C INC,NS)
    DO 780 I=1,MH
    DO 780 J=1,NC
    PRO(I,J)=+PC(J,I)/RC(J,J)
78C DO 790 I=1,MH
    DO 790 J=1,NC
    FBGE(I,J)=C.DO
    DO 790 K=1,MH
    FBGE(I,J)=FBGE(I,J)+GN(I,K)*PRO(K,J)
79C IF (IDSTAB.EQ. 1) GC TO 810
    WRITE (6,I=70)
    CALL RAPRNT (MH,MH,MH,5,GN,4,'(5(1X,1PD12.6))')
    WRITE (6,I=80)
    DO 800 I=1,MH
    X(I,I)=DSQRT(GN(I,I))
80C X(I,I) (6,1E50) (X(I,I),I=1,MH)
    WRITE (6,I=80)
810 WRITE (6,I=80)
    DO 820 I=1,MH
    FBGE(I,J)=FBGE(I,J),J=1,NO)
82C FBGE(I,J) (6,1E40) (FBGE(I,J),J=1,NO)
C-----COMPUTE MODAL K MATRIX OPEN LOOP U-INV SAVED IN WNORMI&-----
    IF (IMODE.I) GO TC 830
    CALL INUE (WNORMI,FBGE,AA,MH,MH,NO,4)
83C CALL INUE
    CCNT INUE
C-----RESET FLAG AND F MATRIX FOR ITERATIVE DESTABILIZATION CASE-----
    IF (IDSTAB.EQ. 0) GC TO 850
    DO 840 I=1,NS
    DO 840 J=1,NS
84C BA(I,J)=BA(I,J)-DSTORE(I,J)
    IR=2
85C CCNT INUE
    DO 870 I=1,NS
    DO 870 J=1,NS
    SUM=0.0
    DO 860 K=1,NC

```

```

OPT111530
OPT111540
OPT111550
OPT111560
OPT111570
OPT111580
OPT111590
OPT111600
OPT111610
OPT111620
OPT111630
OPT111640
OPT111650
OPT111660
OPT111670
OPT111680
OPT111690
OPT111700
OPT111710
OPT111720
OPT111730
OPT111740
OPT111750
OPT111760
OPT111770
OPT111780
OPT111790
OPT111800
OPT111810
OPT111820
OPT111830
OPT111840
OPT111850
OPT111860
OPT111870
OPT111880
OPT111890
OPT111900
OPT111910
OPT111920
OPT111930
OPT111940
OPT111950
OPT111960
OPT111970
OPT111980
OPT111990
OPT112000

```

```

870      SUM=SUM+FBGE(I,K)*HC(K,J)
      PRO(I,J)=BA(I,J)-SUM
      CALL RAPRNT (NS,NS,NS,5,PRO,4,'(5(1X,1PD13.6))')
      IF (IR.LT.2) GO TO 890
      CALL BALANC (NS,NS,PRO,LOW,IHIGH,D1)
      CALL ORTHES (NS,NS,LCW,IHIGH,PRO,D2)
      CALL ORTRAN (NS,NS,LCW,IHIGH,PRO,CR,C1,GM,IERR)
      CALL HQR2 (NS,NS,LOW,IHIGH,PRO,CR,C1,GM,IERR)
      IF (IERR.NE.0) CALL EREXIT (NS,PRO,IERR)
      CALL BALBAK (NS,NS,LCW,IHIGH,D1,NS,GM)
      WRITE (6,1560)
C-----NORMALIZE AND PRINT SUBOPT. ESTIMATOR EIGENSYSTEM-----
      IWRITE=5
      CALL CNORM (CR,C1,GM,NS,IWRITE,NSQ,DDD,D1,D2,WNCRM,WNORMI,HC,AA,
1NO,NS)
      DC 880      I=1,NS
      IF (CR(I).LT.0.0) GO TO 880
      WRITE (5,1660)
      RETURN
880      CCNT INUE
      CC TO 900
890      IF (IQ.EQ.C) GO TO 1260
      DC 910      I=1,NO
      DC 910      J=1,MH
      DO 910      J=0,LC
      PRO(I,J)=0.0
      DO 910      K=1,NC
      PRO(I,J)=PRC(I,J)+RC(I,K)*FBGE(J,K)
      DO 920      I=1,MH
      DO 920      J=1,MH
      CQ(I,J)=0.0
      DO 920      K=1,NC
      CQ(I,J)=CQ(I,J)-FBGE(I,K)*PRO(K,J)
      CCNT INUE
      C-----THE RMS STATE AND CONTROL RESPONSES-----
      IR=IR+1
      GO TO (1090,1090,940,940), IR
      DC 950      I=1,NS
      DC 950      J=1,NG
      DO 950      J=0,0
      X(I,J)=0.0
      DO 950      K=1,NG
      X(I,J)=X(I,J)+GAM(I,K)*Q(K,J)
      DC 970      I=1,NS
      DC 970      J=1,NS
      SUM=0.0
      DO 960      K=1,NG
      SUM=SUM-X(I,K)*GAM(J,K)
      PPD(I,J)=SUM+CQ(I,J)

```



```

          PRO(J,I)=FFC(I,J)
          CQ(I,J)=SUM
          CQ(J,I)=SUM
          W21(I,J)=GM(I,J)
          W21(I,J)=GM(J,I)
          CALL MINV (NSQ,W21,NS,DDD,D1,D2)
          CALL SCOV (NS,GM,W21,CR,CI,NS,GM,W21,CR,CI,PRC,GN)
          WRITE (6,1670)
          CALL RAPRNT (MH,MH,MT,5,GN,4,'(5(1X,1PD13.6))')
          WRITE (6,1680)
          DO 980 I=1,MH
            X(I,I)=DSQRT(GN(I,I))
            WRITE (6,1690) (X(I,I),I=1,MH)
            IF (I*EQ*EQ*CI) GO TO 1260
            DO 1000 I=1,NC
              DO 1000 J=1,NS
                SUM=0.0
                DO 990 K=1,NS
                  SUM=SUM+FBGC(I,K)*GN(K,J)
                X(I,J)=SUM
                DO 1020 I=1,NS
                  DO 1020 J=1,NS
                    SUM=0.0
                    IF (NC*EQ*CI) GO TO 1020
                    DO 1010 K=1,NC
                      SUM=SUM+G(I,K)*X(K,J)
                    PRO(I,J)=CQ(I,J)+SUM
                    CALL SCOV (NS,SC,W11,CWR,CWI,NS,GM,W21,CR,CI,PRO,BA)
                    IF (NC*EQ*CI) GO TO 1040
                    DO 1030 I=1,NC
                      DO 1030 J=1,NS
                        W21(I,J)=0.0
                        DO 1030 K=1,NS
                          W21(I,J)=W21(I,J)+FBGC(I,K)*BA(J,K)
                        DO 1060 I=1,NS
                          DO 1060 J=1,NS
                            SUM=0.0
                            IF (NC*EQ*CI) GO TO 1060
                            DO 1050 K=1,NC
                              SUM=SUM+G(I,K)*W21(K,J)
                              PRO(I,J)=SUM
                              DO 1070 I=1,NS
                                DO 1070 J=1,NS
                                  PRO(I,J)=PRC(I,J)
                                  CALL SCOV (NS,SC,W11,CWR,CWI,NS,SC,W11,CWP,CWI,PRO,CQ)
                                  DO 1080 I=1,NS
                                    DO 1080 J=1,NS

```

```

OPTI12490
OPTI12500
OPTI12510
OPTI12520
OPTI12530
OPTI12540
OPTI12550
OPTI12560
OPTI12570
OPTI12580
OPTI12590
OPTI12600
OPTI12610
OPTI12620
OPTI12630
OPTI12640
OPTI12650
OPTI12660
OPTI12670
OPTI12680
OPTI12690
OPTI12700
OPTI12710
OPTI12720
OPTI12730
OPTI12740
OPTI12750
OPTI12760
OPTI12770
OPTI12780
OPTI12790
OPTI12800
OPTI12810
OPTI12820
OPTI12830
OPTI12840
OPTI12850
OPTI12860
OPTI12870
OPTI12880
OPTI12890
OPTI12900
OPTI12910
OPTI12920
OPTI12930
OPTI12940
OPTI12950
OPTI12960

```



```

11C80 GM(I,J)=CQ(I,J)-BA(I,J)-BA(J,I)+GN(I,J)
11C80 GM(J,I)=GM(I,J)
11C80 GO TO 1100
11C90 CALL SCOV(NS,SC,W11,CWR,CWI,NS,SC,W11,CWR,CWI,CG,GM)
11100 IF (NC.EQ.0) GO TO 1150
11C90 DO 1120 I=1,NS
11C90 DO 1120 J=1,NC
11C90 DO 1120 K=1,NS
11C90 PRO(I,J)=O.CC
11C90 DO 1110 K=1,NS
11C90 PRO(I,J)=PFC(I,J)+GM(I,K)*FBGC(J,K)
11110 CONTINUE
11120 DO 1140 I=1,NC
11120 DO 1140 J=1,NC
11120 SC(I,J)=O.CC
11130 DO 1130 K=1,NS
11140 SC(I,J)=SC(I,J)+FBGC(I,K)*PRO(K,J)
11150 CONTINUE
11150 IF (IREG.EC.0) GO TO 1170
11160 DO 1160 I=1,NS
11160 DO 1160 J=1,NS
11160 GM(I,J)=GM(I,J)
11170 GO TO 1190
11170 WRITE(6,1700)
11170 CALL RAPRNT(MH,MH,MF,5,GM,4,(5(1X,1PD13.6)))
11170 IF (IR.GT.2) GO TO 1190
11180 DO 1180 I=1,MH
11180 DO 1180 J=1,MH
11180 CQ(I,J)=GN(I,J)+GM(I,J)
11190 CONTINUE
11190 WRITE(6,1710)
11190 CALL RAPRNT(MH,MH,MF,5,CQ,4,(5(1X,1PD13.6)))
11190 IF (NC.EQ.0) GO TO 1210
11200 DO 1200 I=1,NC
11210 DO 1200 J=1,NS
11220 CQ(I,J)=DSGT(CQ(I,I))
11220 IF (NC.EQ.0) GO TO 1240
11230 DO 1230 I=1,NC
11240 SC(I,J)=DSGT(SC(I,I))
11240 WRITE(6,1740)
11240 DO 1250 I=1,NS
11240 IF (I.LT.NC) WRITE(6,1750) CQ(I,I),SC(I,I)
11250 IF (I.GT.NC) WRITE(6,1750) CQ(I,I)
11260 CONTINUE
11260 IF (ITF3.EC.0) GO TO 1290
11260 FORM COMPENSATOR FROM MEAS TO INPUT AND COMPUTE TF-----
11260 DC 1280 I=1,NS

```

```

1270 DO 1280 J=1,NS
1280 SUM=0
DO 1270 K=1,NC
SUM=SUM+FBGE(I,K)*HO(K,J)
CQ(I,J)=ACL(I,J)-SUM
WRITE(6,1760)
ITFX=3
IZERO=0
CALL TF (NS,NS,NSQ,CQ,AA,NC,FBGE,BM,NC,FBGC,CM,IZERC,D,BB,CC,CP,
1WR,WI,CWR,CWI,SC,JCF,RES,D1,D2,DDD,EPS,ITF3,ITFX)
1290 CCONTINUE
C----- COMPUTE PSD FUNCTIONS OF THE CONTROLLED SYSTEM -----
IF (IPSD.EC.O) GO TO 1310
IF (IYU.LT.3) GO TO 1300
CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HU,HQ,FBGE,NG,
1GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,1,IPSD,INORM)
1CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HY,HC,FBGE,NG,
1GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,2,IPSD,INORM)
GO TO 1310
1200 CALL PSDCAL (M,NS,RM,X,NC,GW,GV,FBGC,NO,HY,HU,HC,FBGE,NG,
1GAM,ACL,BA,WR,WI,D1,D2,JCF,RES,Q,RC,BB,CC,IYU,IPSD,INORM)
1310 IF (IS.EC.O) RETURN
IF (INC.NE.O) GO TO 1330
DO 1320 I=1,NS
DO 1320 J=1,NS
ACL(I,J)=BA(I,J)
1320 CCONTINUE
CALL MINV (NSQ,ACL,NS,DDD,D1,D2)
CALL READW (NG,WR)
WRITE(6,1770) (WR(I),I=1,NG)
1340 DO 1340 I=1,NS
WI(I)=0
DO 1340 J=1,NG
WI(I)=WI(I)+GAM(I,J)*WR(J)
134C DO 1360 I=1,NS
DO 1360 J=1,NS
CR(I)=0
DO 1350 J=1,NS
CR(I)=CR(I)-ACL(I,J)*WI(J)
1350 DO 1360 I=1,NS
DO 1360 J=1,NS
CR(I)=CR(I)+FBGC(I,J)*CR(J)
1370 WRITE(6,1780) (CR(I),I=1,NC)
RETURN
C-----
C67C FORMAT (2X,1P6D14.6,/,2X,6D14.6)

```



```

1770 FORMAT (//,EX,46HSTEADY DISTURBANCE VECTOR.....W...//OPT114410
1,10(1X,1PD12.4//))OPT114420
1780 FORMAT (//,EX,45HSTEADY STATE VALUES OF STATE VAR. ARE.....//OPT114430
1790 FORMAT (//,EX,47HSTEADY STATE CONTROL IS .....//OPT114440
1,10(1X,1PD12.4//))OPT114450
1800 FORMAT (//,EX,49HENTER THE MAGNITUDE OF THE DESTABILIZATION VECTOROPT114460
1,/.8X,47HTC BE ADDED DOWN THE DIAGONAL OF THE "F"-MATRIX,/.8X,18HTOPT114470
2C DESTABILIZE IT,//)OPT114480
C-----CFGRAPH DATA-----OPT114490
2C00 FORMAT(5X,2C30.14)OPT114500
C-----OPT114510
C=====OPT114520
ENDOPT114530
SUBROUTINE FAPRNT (NMAX,M,N,L,A,IDIM,FMT)OPT114540
REAL*8 A(NMAX,N)OPT114550
DIMENSION FMT(IDIM)OPT114560
NU=LOPT114570
DO 20 NL=1,N,LOPT114580
IF (NU.GT.N) NU=NOPT114590
DC 10 I=1,NOPT114600
WRITE (6,FMT) (A(I,J),J=NL,NU)OPT114610
NU=NU+LOPT114620
RETURNOPT114630
FORMAT (1X)OPT114640
ENDOPT114650
C=====OPT114660
SUBROUTINE RGAIN (M,NS,NC,NOB,WR,WI,VF,GN,W1,TCB,W21,LT,C,CI,CT,M
1HS,MT)OPT114670
IMPLICIT REAL*8 (A-H,O-Z)OPT114680
DIMENSION WR(M),WI(M),VF(M,M),GN(NS,NS)OPT114690
DIMENSION W1(NS,NS),TCB(M,M),W21(NS,NS),LT(NS),MT(NS)OPT114700
DIMENSION C(NS),CI(NS),CT(NS,NS)OPT114710
K=1OPT114720
KP=1OPT114730
KN=1OPT114740
NRZEV=0OPT114750
NCPZEV=0OPT114760
IF (K.GT.M) GO TO 21C OPT114770
C-----OPT114780
C CHECK FOR EIGVAL AT OR NEAR J-OMEGA AXIS TO INCLUDE IN E-L FIGSYS OPT114800
C TURN FIRST ONE POSITIVE AND SECOND ONE NEGATIVE OPT114810
C-----OPT114820
FIGVR=CABS(WR(K))OPT114830
IF (EIGVR.GE.1.D-10) GO TO 60 OPT114840
IF (WI(K)) 40,20,40 OPT114850
NRZEV=NRZEV+1 OPT114860
IF (NRZEV.GT.1) GO TO 30 OPT114870
OPT114880

```

```

30      WR(K)=EIGVP
      GO TO 80
      WR(K)=-EIGVF
      WRITE (6,25C)
      GO TO 150
40      NCPZEV=NCPZEV+1
      IF (NCPZEV.GT.1) GO TO 50
      WR(K)=EIGVR
      WR(K+1)=EIGVR
      GO TO 110
50      WR(K)=-EIGVF
      WR(K+1)=-EIGVR
      WRITE (6,30C)
      GO TO 180
60      IF (WR(K)) 140,70,70
70      IF (WI(K)) 110,80,110
      C-----EIGENVECTCR FOR REAL EIGENVALUE, POSITIVE-----
80      IF (NOB.EQ.C) GO TO 100
      DO 90 J=1,M
      TCBC(J,KP)=VF(J,K)
      KP=KP+1
      K=K+1
90      GO TO 10
100     C-----EIGENVECTCR FOR COMPLEX EIGENVALUE, PCSITIVE REAL PART-----
110     IF (NOB.EQ.C) GO TO 130
      DO 120 J=1,M
      FR=VF(J,K)
      FI=-VF(J,K+1)
      TCBC(J,KP)=FR+FI
      TCBC(J,KP+1)=FR-FI
      KP=KP+2
      K=K+2
120     GO TO 10
130     IF (WI(K)) 180,150,180
      C-----EIGENVECTCR FOR REAL EIGENVALUE, NEGATIVE REAL PART-----
140     C(KN)=WR(K)
150     C(KN)=WI(K)
      IF (NOB.NE.C) GO TO 170
      KNS=KN+NS
      DO 160 J=1,M
      TCBC(J,KNS)=VF(J,K)
      KN=KN+1
      K=K+1
160     GO TO 10
170     RR=WR(K)
      RI=WI(K)
      C(KN)=RR
      C-----EIGENVECTCR FOR COMPLEX EIGENVALUE, NEGATIVE REAL PART-----
180

```



```

C(KN+1)=RR
CI(KN)=RI
CI(KN+1)=-RI
IF (NOB.NE.0) GO TO 200
KNS=KN+NS
DO 190 J=1,M
FR=VF(J,K)
FI=-VF(J,K+1)
TCB(J,KNS)=FR+FI
TCB(J,KNS+1)=FR-FI
KN=KN+2
K=K+2
GO TO 10
210 CONTINUE
IF (NOB.NE.0) GO TO 240
C-----FORMATION OF W11-----
DO 220 I=1,NS
DO 220 J=1,NS
W11(I,J)=TCB(I,J+NS)
CT(I,J)=W11(I,J)
C-----FORMATION OF W21-----
DO 230 I=1,NS
DO 230 J=1,NS
W21(I,J)=TCB(I+NS,J+NS)
IF (NOB.EQ.0) GO TO 260
DO 250 I=1,NS
DO 250 J=1,NS
W21(I,J)=-TCB(I,J)
W11(I,J)=TCB(I+NS,J)
CONTINUE
C-----INVERT W11-----
NSQ=NS*NS
CALL MINV (NSQ,W11,NS,DEIC,LI,MT)
C-----CALCULATE THE RGAIN MATRIX-----
DO 270 IL=1,NS
DO 270 JL=1,NS
GN(IL,JL)=0.D0
DO 270 KL=1,NS
GN(IL,JL)=GN(IL,JL)+W21(IL,KL)*W11(KL,JL)
IF (NOB.EQ.0) RETURN
DO 280 I=1,NS
DO 280 J=1,NS
CT(I,J)=W11(J,I)
RETURN
C-----EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT, 400PT15810
25C FORMAT (IX,51H EULER-LAGRANGE EQUATIONS HAVE A REAL EIGENVALUE AT, 400PT15810
114H OR NEAR ZERO./)
300 FORMAT (IX,49H EULER-LAGRANGE EQUATIONS HAVE A COMPLEX PAIR OF ,400PT15840

```

```

1EIGENVALUES AT OR NEAR THE J-OMEGA AXIS.)
C=====
END
SUBROUTINE MINV (NSQ,A,N,D,L,M)
IMPLICIT REAL*8 (A-H,C-Z)
DIMENSION A(NSQ),L(N),M(N)
DOUBLE PRECISION A,D,BIGA,HOLD
NM=N*N
L=1.0D0
AK=-N
DO 180 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
CC 20 I=K,N
IJ=IZ+I
IF (DABS(BIGA)-DABS(A(IJ))) 10,20,20
1C BIGA=A(IJ)
L(K)=I
M(K)=J
2C CONTINUE
-----INTERCHANGE ROWS-----
J=L(K)
IF (J-K) 5C,50,30
KI=K-N
DO 40 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
A(JI)=HOLD
4C -----INTERCHANGE COLUMNS-----
I=M(K)
IF (I-K) 8C,80,60
JP=N*(I-1)
DO 70 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
A(JI)=HOLD
7C -----DIVIDE COLUMN BY MINUS PIVOT-----
C----- (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)-----
8C IF (BIGA) 1C0,90,1C0
OPTI 5850
OPTI 5860
OPTI 5870
OPTI 5880
OPTI 5890
OPTI 5900
OPTI 5910
OPTI 5920
OPTI 5930
OPTI 5940
OPTI 5950
OPTI 5960
OPTI 5970
OPTI 5980
OPTI 5990
OPTI 6000
OPTI 6010
OPTI 6020
OPTI 6030
OPTI 6040
OPTI 6050
OPTI 6060
OPTI 6070
OPTI 6080
OPTI 6090
OPTI 6100
OPTI 6110
OPTI 6120
OPTI 6130
OPTI 6140
OPTI 6150
OPTI 6160
OPTI 6170
OPTI 6180
OPTI 6190
OPTI 6200
OPTI 6210
OPTI 6220
OPTI 6230
OPTI 6240
OPTI 6250
OPTI 6260
OPTI 6270
OPTI 6280
OPTI 6290
OPTI 6300
OPTI 6310
OPTI 6320

```

```

9C      D=0.0D0
10C     RETURN
11C     DO 120 I=1,N
12C     IF (I-K) 11C,120,110
13C     IK=NK+I
14C     A(IK)=A(IK)/(-BIGA)
15C     CONTINUE
16C     -----REDUCE MATRIX-----
17C     DO 150 I=1,N
18C     IK=NK+I
19C     HOLD=A(IK)
20C     IJ=I-N
21C     DO 150 J=1,N
22C     IJ=IJ+N
23C     IF (I-K) 130,150,130
24C     IF (J-K) 14C,150,140
25C     KJ=IJ-I+K
26C     A(IJ)=HOLD*A(KJ)+A(IJ)
27C     CONTINUE
28C     -----DIVIDE ROW BY PIVOT-----
29C     KJ=K-N
30C     DO 170 J=1,N
31C     KJ=KJ+N
32C     IF (J-K) 16C,170,16C
33C     A(KJ)=A(KJ)/BIGA
34C     CONTINUE
35C     D=D*BIG
36C     -----PRODUCT OF PIVOTS-----
37C     A(KK)=(1.0D0)/BIGA
38C     CONTINUE
39C     -----REPLACE PIVOT BY RECIPROCAL-----
40C     -----FINAL ROW AND COLUMN INTERCHANGE-----
41C     K=N
42C     K=(K-1)
43C     IF (K) 260,260,200
44C     I=L(K)
45C     IF (I-K) 230,230,210
46C     JQ=N*(K-1)
47C     JR=N*(I-1)
48C     DO 220 J=1,N
49C     JK=JQ+J
50C     HOLD=A(JK)
51C     JI=JR+J
52C     A(JK)=-A(JI)
53C     A(JI)=HOLD
54C     J=M(K)
55C     IF (J-K) 15C,190,240
56C     KI=K-N

```

OPT16330
 OPT16340
 OPT16350
 OPT16360
 OPT16370
 OPT16380
 OPT16390
 OPT16400
 OPT16410
 OPT16420
 OPT16430
 OPT16440
 OPT16450
 OPT16460
 OPT16470
 OPT16480
 OPT16490
 OPT16500
 OPT16510
 OPT16520
 OPT16530
 OPT16540
 OPT16550
 OPT16560
 OPT16570
 OPT16580
 OPT16590
 OPT16600
 OPT16610
 OPT16620
 OPT16630
 OPT16640
 OPT16650
 OPT16660
 OPT16670
 OPT16680
 OPT16690
 OPT16700
 OPT16710
 OPT16720
 OPT16730
 OPT16740
 OPT16750
 OPT16760
 OPT16770
 OPT16780
 OPT16790
 OPT16800

```

DO 250 I=1,N
KI=KI+N
HOLD=A(KI)
JI=KI-K+J
A(KI)=-A(JI)
A(JI)=HOLD
GO TO 190
250 K=0
260 RETURN
END
C=====
SUBROUTINE SCOV (NL,KL,WLI,VL1,VL2,NR,WR,WRI,VR1,VR2,Q,X)
REAL*8 VLI(NL),VL2(NL),WL(NL,NL),WLI(NL,NL),X(NL,NR),Q(NL,NR),
1 VRI(NR),VR2(NR),WR(NR,NR),WRI(NR,NR)
1 REAL*8 A,B,C,D,K1,K2,K3,K4
DO 20 I=1,NL
DO 20 J=1,NR
X(I,J)=0.
DO 20 II=1,NL
X(I,J)=X(I,J)+WLI(I,II)*Q(II,J)
DO 40 I=1,NL
DO 40 J=1,NR
Q(I,J)=0.
DO 30 JJ=1,NR
Q(I,J)=Q(I,J)+X(I,JJ)*WRI(J,JJ)
CGNT INUE
I=1
IF (VL2(I)) 60,110,60
J=1
IF (VR2(J)) 80,90,80
A=VLI(I)+VRI(J)
B=-2.*VL2(I)*VR2(J)
C=A*2+VL2(I)*2+VR2(J)*2
L=C*2-B*2
K1=A*C/D
K2=-(VR2(J)*C+VL2(I)*B)/D
K3=-(VR2(J)*B+VL2(I)*C)/D
K4=-A*B/D
II=I+1
JI=J+1
X(II,JI)=+K1*Q(I,J)+K2*Q(I,J1)+K3*Q(II,J)+K4*Q(II,J1)
X(II,J1)=-K2*Q(I,J)+K1*Q(II,J)-K4*Q(II,J1)
X(II,J)=-K3*Q(I,J1)+K1*Q(II,J)+K2*Q(II,J1)
X(II,J1)=+K4*Q(I,J)-K3*Q(II,J1)-K2*Q(II,J1)
J=J+2
GO TO 100
90 A=VRI(J)+VLI(I)
B=A*2+VL2(I)*2

```

```

100      K1=A/B
      K2=VL2(I)/B
      X(I,J)=K1*C(I,J)-K2*C(I+1,J)
      X(I+1,J)=K2*Q(I,J)+K1*Q(I+1,J)
      J=J+1
      IF (J.LE.NR) GO TO 70
      I=I+2
      GO TO 160
110      J=1
120      IF (VR2(J)) 120,140,130
130      A=VR1(J)+VL1(I)
      B=A**2+VR2(J)**2
      K1=A/B
      K2=VR2(J)/B
      X(I,J)=K1*C(I,J)-K2*C(I,J+1)
      X(I,J+1)=K2*Q(I,J)+K1*Q(I,J+1)
      J=J+2
      GO TO 150
140      X(I,J)=Q(I,J)/(VR1(J)+VL1(I))
      J=J+1
      IF (J.LE.NR) GO TO 120
150      I=I+1
      IF (I.LE.NL) GO TO 50
      DO 170 I=1,NL
      CO 170 J=1,NR
      Q(I,J)=0.
      DO 170 I=1,NL
      Q(I,J)=Q(I,J)+WL(I,I)*X(I,I,J)
      DO 190 I=1,NL
      CO 190 J=1,NR
      X(I,J)=0.
      DO 180 JJ=1,NR
      X(I,J)=X(I,J)+Q(I,JJ)*WR(J,JJ)
      CCNT=INUE
      RETURN
160      END
170
180
190
=====
C=====SUBROUTINE MODE (WNORM,G,GNORM,NS,N1,N2,ICCN)
C      WNORM TRANSFORMATION MATRIX U OR U-INV
C      NS NO. OF STATE
C      NC NO. OF INPUTS CR OUTPUTS
C      ICCN CONTROL FLAG TO INDICATE WHICH TRANSFORMATION
C      Q = MCLAL GAMMA
C      1 = MCLAL H
C      2 = MCLAL C
C      3 = MCLAL K
C      4 = MCLAL K
=====
OPTI17290
OPTI17300
OPTI17310
OPTI17320
OPTI17330
OPTI17340
OPTI17350
OPTI17360
OPTI17370
OPTI17380
OPTI17390
OPTI17400
OPTI17410
OPTI17420
OPTI17430
OPTI17440
OPTI17450
OPTI17460
OPTI17470
OPTI17480
OPTI17490
OPTI17500
OPTI17510
OPTI17520
OPTI17530
OPTI17540
OPTI17550
OPTI17560
OPTI17570
OPTI17580
OPTI17590
OPTI17600
OPTI17610
OPTI17620
OPTI17630
OPTI17640
OPTI17650
OPTI17660
OPTI17670
OPTI17680
OPTI17690
OPTI17700
OPTI17710
OPTI17720
OPTI17730
OPTI17740
OPTI17750
OPTI17760

```



```

C      5 = CONTROL EIGENVECTOR MATRIX
C      6 = MEASUREMENT EIGENVECTOR MATRIX
=====
IMPLICIT REAL*8(A-H,C-Z)
DIMENSION WNORM(NS,NS),G(N1,N2),GNORM(N1,N2)
DO 10 I=1,N1
DO 10 J=1,N2
  GNORM(I,J)=C.
  IPOINT=ICONT+1
GO TO (20,2C,50,90,2C,90,90), IPCINT
DO 30 J=1,N2
DO 30 I=1,N1
  GNORM(I,J)=GNORM(I,J)+WNORM(I,K)*G(K,J)
GO TO (40,7C,90,90,80), IPCINT
WRITE (6,17C)
DO 60 I=1,NS
  RETURN (GNORM(I,J),J=1,N2)
WRITE (6,18C)
GO TO 50
WRITE (6,24C)
GO TO 50
DO 100 J=1,NS
DO 100 I=1,N1
DO 100 K=1,NS
  GNORM(I,J)=GNORM(I,K)*WNORM(K,J)
GO TO (110,110,110,120,110,130,140), IPOINT
WRITE (6,19C)
GO TO 150
WRITE (6,20C)
GO TO 150
WRITE (6,21C)
GO TO 150
WRITE (6,22C)
DO 160 I=1,N1
WRITE (6,23C) (GNORM(I,J),J=1,NS)
RETURN
=====
FORMAT (//,EX,45HMODAL CONTROL DISTRIBUTION MATRIX.....TI*G.....//) OPT117770
FORMAT (//,EX,50HMODAL PROCESS NOISE DISTRIBUTION MATRIX.....TI*GAM.....//) OPT117780
1. FORMAT (//) OPT117790
FORMAT (//,EX,45HMODAL MEASUREMENT SCALING MATRIX.....H(BAR)*T.....//) OPT117800
FORMAT (//,EX,45HMODAL MODAL CONTROL GAINS.....C*T.....//) OPT117810
FORMAT (//,EX,45HMODAL EIGENVECTOR MATRIX.....C*M.....//) OPT117820
FORMAT (//,EX,45HMODAL MEASUREMENT EIGENVECTOR MATRIX.....H(BAR)*M.....//) OPT117830
FORMAT (1X,(2X,1P6D14.6)) OPT117840
FORMAT (//,EX,45HMODAL FILTER STEADY STATE GAINS.....TI*K.....//) OPT117850
OPT117860
OPT117870
OPT117880
OPT117890
OPT117900
OPT117910
OPT117920
OPT117930
OPT117940
OPT117950
OPT117960
OPT117970
OPT117980
OPT117990
OPT118000
OPT118010
OPT118020
OPT118030
OPT118040
OPT118050
OPT118060
OPT118070
OPT118080
OPT118090
OPT118100
OPT118110
OPT118120
OPT118130
OPT118140
OPT118150
OPT118160
OPT118170
OPT118180
OPT118190
OPT118200
OPT118210
OPT118220
OPT118230
OPT118240

```



```

      3C      VECIN=(-VR*VMI+VI*VMP)/EMAX
      WNCRN(I,K)=VECRN
      WNCRN(I,K+1)=VECIN
      CCNT INUE
      KK=1
      GO TO 50
      4C      KK=0
      5C      CCNT INUE
      -----NORMALIZE REAL EIGENVECTORS BY THE TOTAL LENGTH-----
      DO 80 K=1,NS
      IF (DABS(WY(K)).GE.1.D-10) GO TO 80
      LR=LR+1
      REMOD=0.D0
      DO 60 I=1,NS
      REMOD=VEC(I,K)*2+REMOD
      REMOD=DSQRT(REMOD)
      DO 70 I=1,NS
      RVEC=VEC(I,K)/REMOD
      WNCRN(I,K)=RVEC
      CCNT INUE
      70      CCNT INUE
      80      GO TO (90,100,110,120,130), IWRITE
      9C      WRITE (6,32C)
      10C      GO TO 140
      110      WRITE (6,33C)
      120      WRITE (6,34C)
      130      WRITE (6,35C)
      140      WRITE (6,36C)
      KK=0
      NPRTW=0
      NFMTW=1
      DO 180 I=1,NS
      IF (KK.EQ.1) GO TO 170
      IF (DABS(WY(I)).GT.1.D-10) KK=1
      IF (DABS(WY(I)).GT.1.D-10) KK=1
      PRINT OUT NC MORE THAN 6 WORDS, NOT SEPARATING CCMPLEX EIGVAL
      IF (NPRTW.LT.5) OR (NPRTW.EQ.5.AND.KK.EQ.0) GC TO 150
      FMT(NFMTW+1)=RIGHT
      WRITE (6,FMT) (STORE(J),J=1,NPRTW)
      NPRTW=0
      NFMTW=1
      NPRTW=1
      NFMTW=NPRTW+1
      NPRTW=NPRTW+1
      IF (KK.EQ.1) GO TO 160
      STORE(NPRTW)=WZ(I)
      FMT(NFMTW)=FIELD

```

160	NFMTW=NFMTW+1	OPTI 9210
	FMT(NFMTW)=SEMCOL	OPTI 9220
	GO TO 180	OPTI 9230
	STORE(NPRTW)=WZ(I)	OPTI 9240
	FMT(NFMTW+1)=FIELD	OPTI 9250
	FMT(NFMTW+1)=COMMA	OPTI 9260
	STORE(NPRTW+1)=WY(I)	OPTI 9270
	FMT(NFMTW+2)=FIELD	OPTI 9280
	FMT(NFMTW+3)=SEMCOL	OPTI 9290
	NFMTW=NFMTW+3	OPTI 9300
	NPRTW=NPRTW+1	OPTI 9310
	GO TO 180	OPTI 9320
170	KK=0	OPTI 9330
180	CCNT INUE	OPTI 9340
	FMT(NFMTW)=SEMENT	OPTI 9350
	FMT(NFMTW+1)=RIGHT	OPTI 9360
	WRITE(6,FMT) (STORE(J),J=1,NPRTW)	OPTI 9370
	IF (IWRITE.EQ.1) GO TO 190	OPTI 9380
	WRITE(6,37C)	OPTI 9390
	GO TO 200	OPTI 9400
190	WRITE(6,38C)	OPTI 9410
200	CALL RAPRNT (NS,NS,NS,6,WNORM,4,(6(1X,1PD13.6)))	OPTI 9420
210	GO TO 230	OPTI 9430
	CALL MCDE (WNORM,H0,CM,NS,N1,N2,5)	OPTI 9440
	GO TO 230	OPTI 9450
220	CALL MCDE (WNORM,H0,CM,NS,N1,N2,6)	OPTI 9460
230	GO TO 240	OPTI 9470
240	WRITE(6,35C)	OPTI 9480
	GO TO 290	OPTI 9490
250	WRITE(6,40C)	OPTI 9500
	GO TO 290	OPTI 9510
260	WRITE(6,41C)	OPTI 9520
	GO TO 290	OPTI 9530
270	WRITE(6,420)	OPTI 9540
	GO TO 290	OPTI 9550
280	WRITE(6,43C)	OPTI 9560
C----	SAVE U-INVERSE OPEN LOOP IN WNORMI	OPTI 9570
29C	IF (IWRITE.GT.1) GC TO 310	OPTI 9580
	DO 300 I=1,NS	OPTI 9590
	DO 300 J=1,NS	OPTI 9600
30C	WNORMI(I,J)=WNORM(I,J)	OPTI 9610
	CALL MINV (ASQ,WNORMI,NS,DDD,D1,D2)	OPTI 9620
	CALL RAPRNT (NS,NS,NS,6,WNORMI,4,(6(1X,1PD13.6)))	OPTI 9630
	RETURN	OPTI 9640
31C	CALL MINV (ASQ,WNORM,NS,DDD,D1,D2)	OPTI 9650
	CALL RAPRNT (NS,NS,NS,6,WNORM,4,(6(1X,1PD13.6)))	OPTI 9660
C----	RETURN	OPTI 9670
		OPTI 9680


```

C=====
END
SUBROUTINE POLES (N,NM,A,AA,M,B,L,C,EVR,EVI,D1,D2,JCF,SC)
IMPLICIT REAL*8(A-H,C-Z)
DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),EVR(N),EVI(N),D1(N),D2(N),JCF(N),SC(N,N)
DO 10 I=1,N
DO 10 J=1,N
AA(I,J)=A(I,J)
CALL BALANC (NM,N,AA,LGW,IHIGH,D1)
CALL ORTHES (NM,N,LGW,IHIGH,AA,D2)
CALL ORTRAN (NM,N,LGW,IHIGH,AA,D2,SC)
CALL HQR2 (NM,N,LGW,IHIGH,AA,EVR,EVI,SC,IERR)
IF (IERR.NE.0) GO TO 30
CALL BALBAC (NM,N,LGW,IHIGH,D1,N,SC)
WRITE (6,4C)
DC 20 I=1,N
C-----OPTGRCM DATA-----
WRITE (4,2CC) EVR(I),EVI(I)
C-----
2C WRITE (6,5C) EVR(I),EVI(I)
3C RETURN
4C RETURN
C-----
40 FORMAT (//,28H TF DENOMINATOR EIGENVALUES:,/ )
50 FORMAT (//,2X,3H (,F13.6,4H)+J(,F13.6,1H))
60 FORMAT (//,35H FAILURE IN HQR2, CALCULATING PCLES)
C-----OPTGRCM DATA-----
2000 FORMAT(5X,2C30.14)
C-----
END
C=====
SUBROUTINE ZEROS (K1,K2,IFDWF,N,NM,A,AA,M,E,L,C,D,BB,CC,CP,EVR,EVI,
1 D1,D2,EPS,ITFX)
IMPLICIT REAL*8(A-H,C-Z)
DIMENSION A(N,N),AA(N,N),B(N,M),C(L,N),D(L,M),BE(N),CC(N),CP(N),EVR(N),EVI(N),D1(N),D2(N)
DOUBLE PRECISION SCL,DABS
IF (ITFX.EQ.1) IFL = 10
IF (ITFX.EQ.2) IFL = 1
IF (ITFX.EQ.3) IFL = 4
DC 10 I=1,N
EB(I)=B(I,K1)
CC(I)=C(K2,I)
DO 10 J=1,N

```

1C	AA(I,J)=A(I,J)	OPT20650
	WRITE (6,90) K1,K2	OPT20660
	IF (IFCFW.EQ.0) GO TO 20	OPT20670
	H=D(K2,K1)	OPT20680
	IF (DABS(H).LE.EPS) GO TO 20	OPT20690
	JJ=N	OPT20700
	GO TO 50	OPT20710
20	NN=N-1	OPT20720
	DO 30 I=1,NN	OPT20730
	H=SCL(N,BB,CC)	OPT20740
	CALL CCOMP (N,NM,AA,CC,CP)	OPT20750
	IF (DABS(H).GT.EPS) GO TO 40	OPT20760
30	CONTINUE	OPT20770
	F=SCL(N,BB,CC)	OPT20780
	WRITE (6,100) H	OPT20790
C	-----OPGRAPH DATA FOR "0" CRDER-----	OPT20800
	KK=ITFX	OPT20810
	WRITE (IFL,300) KK,K2,K1	OPT20820
	ORDER = C.O	OPT20830
	WRITE (IFL,301) CRDER,H	OPT20840
C	-----	OPT20850
	GO TO 70	OPT20860
40	JJ=N-1	OPT20870
50	WRITE (6,110) JJ,H	OPT20880
C	-----OPGRAPH DATA -----	OPT20890
	KK=ITFX	OPT20900
	WRITE (IFL,300) KK,K2,K1	OPT20910
	ORDER = FLOAT(JJ)	OPT20920
	WRITE (IFL,301) CRDER,H	OPT20930
C	-----	OPT20940
	CALL ACOMP (N,NM,AA,BB,CC,H)	OPT20950
	CALL BALANC (NM,N,AA,LOW,HIGH,D1)	OPT20960
	CALL ORTHES (NM,N,LOW,HIGH,AA,D2)	OPT20970
	CALL HQR (NM,N,LOW,HIGH,AA,EVR,EVI,IERR)	OPT20980
	IF (IERR.NE.0) GO TO 80	OPT20990
	WRITE (6,120)	OPT21000
	DO 60 I=1,N	OPT21010
60	WRITE (6,130) EVR(I),EVI(I)	OPT21020
C	-----OPGRAPH DATA -----	OPT21030
	WRITE (IFL,302) K2,K1	OPT21040
	DC 53 LL = 1,N	OPT21050
63	WRITE (IFL,301) EVR(LL),EVI(LL)	OPT21060
C	-----	OPT21070
70	RETURN	OPT21080
80	WRITE (5,140)	OPT21090
	RETURN	OPT21100
90	FORMAT (///,17H TF FOR INPUT NO.,I3,15H AND CUTPUT NO.,I3,1H:)	OPT21110
		OPT21120

```

100 FORMAT (//,5X,27HNO FINITE ZEROS, TF GAIN =,E12.4)
110 FORMAT (//,3X,20HORDER OF NUMERATOR =,I3.9X,9HIF GAIN =,E12.4)
120 FORMAT (//,3X,57HNUMERATOR EIGENVALUES (INCLUDING EXTRANEOUS ZERO
121 VALUES))
120 FCRMAT (//,4X,1H,(F13.6,4H)+J,(F13.6,1H))
140 FORMAT (52H FAILURE IN HQR CALCULATING TRANSFER FUNCTION ZEROS)
C-----OPGRAPH DATA-----
300 FORMAT (5X,315)
301 FCRMAT (5X,2130.14)
302 FORMAT (5X,215)
C-----
C=====
SUBROUTINE ACOMP (N,NM,A,B,C,F)
REAL*8 A,B,C,F
DIMENSION A(NM,N),B(N),C(N)
DO 10 I=1,N
DO 10 J=1,N
A(I,J)=A(I,J)-B(I)*C(J)/H
RETURN
END
10
C=====
SUBROUTINE CCOMP (N,NM,A,C,CC)
REAL*8 A,C,CC
DIMENSION A(NM,N),C(N),CC(N)
DO 10 I=1,N
CC(I)=0.
DO 10 J=1,N
CC(I)=CC(I)+C(J)*A(J,I)
DO 20 I=1,N
C(I)=CC(I)
RETURN
END
10
20
C=====
FUNCTION SCL (N,B,C)
REAL*8 B,C,SCL
DIMENSION B(N),C(N)
SCL=0.
DO 10 I=1,N
SCL=SCL+C(I)*B(I)
RETURN
END
10
C=====
SUBROUTINE RESID (K1,K2,N,JCF,M,BM,L,CM,PR,PI,RES,BB,CC,IPT)
IMPLICIT REAL*8(A-H,C-Z)
DIMENSION JCF(N),BM(N,M),CM(L,N),PR(N),PI(N),RES(N),BB(N),CC(N),PR
17(4)
DATA SN/8H* SIN(B*T)/,R1/8H */,R2/8HEXP(A*T)/,SD/1H)/

```

```

DATA ZERO/C,DO/,T1/4H*T**/,BLANK/8H      /,CS/8H* COS(B*T/
C-----TEMPORARY MCD TILL JCF IS CALCULATED-----OPT211610
DC 10 I=1,N-----OPT211620
JCF(I)=0-----OPT211630
C-----TEMPORARY MOD-----OPT211640
IF (IPT.EQ. 1) WRITE (6,170)-----OPT211650
CO 20 I=1,N-----OPT211660
BB(I)=BM(I,K1)-----OPT211670
CC(I)=CM(K2,I)-----OPT211680
C-----LOOP THROUGH THE POLES-----OPT211690
I=0-----OPT211700
I=I+1-----OPT211710
IF (I.EQ. 1) GO TO 160-----OPT211720
IF (JCF(I).EQ. 1) GO TO 60-----OPT211730
IF (DABS(PI(I)).LT. 1.D-10) GO TO 50-----OPT211740
C-----COMPUTE SIMPLE POLE RESIDUES AND PRINT BOTH-----OPT211750
RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)-----OPT211760
RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I)-----OPT211770
IF (IPT.EQ. 0) GO TC 40-----OPT211780
PRT(1)=BLANK-----OPT211790
PRT(2)=R2-----OPT211800
IF (PI(I).EQ. 0.D0) PRT(2)=BLANK-----OPT211810
PRT(3)=CS-----OPT211820
PRT(4)=ED-----OPT211830
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)-----OPT211840
I=I+1-----OPT211850
PRT(2)=SN-----OPT211860
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)-----OPT211870
GO TO 30-----OPT211880
I=I+1-----OPT211890
GO TO 30-----OPT211900
C-----COMPUTE SIMPLE REAL POLE RESIDUE-----OPT211910
RES(I)=CC(I)*BB(I)-----OPT211920
IF (IPT.EQ. 0) GO TC 30-----OPT211930
PRT(1)=R1-----OPT211940
PRT(2)=R2-----OPT211950
PRT(3)=BLANK-----OPT211960
PRT(4)=BLANK-----OPT211970
WRITE (6,180) PR(I),PI(I),RES(I),(PRT(J),J=1,4)-----OPT211980
GO TO 30-----OPT211990
C-----LOOK AHEAD TO DETERMINE SIZE OF THE JCRCAN BLOCK-----OPT220000
KT=1-----OPT220010
KT=N-I-----OPT220020
DO 70 J=I,KT-----OPT220030
IF (JCF(J).EQ. 0) GO TO 80-----OPT220040
K=K+1-----OPT220050
C-----OPT220060
C-----OPT220070
C-----OPT220080

```



```

C----- IF (DABS(PI(I)) .LT. 1.D-10) GO TO 110
C----- COMPUTE REPEATED COMPLEX POLE AND PRINT CUT ALL FOUR-----
K=1
RES(I)=CC(I)*BB(I)+CC(I+1)*BB(I+1)+CC(I+2)*BB(I+2)+CC(I+3)*BB(I+3)
RES(I+1)=CC(I)*BB(I+1)-CC(I+1)*BB(I)+CC(I+2)*BB(I+3)-CC(I+3)*BB(I+4)
12 RES(I+2)=CC(I)*BB(I+3)+CC(I+1)*BB(I+2)
RES(I+3)=CC(I)*BB(I+3)-CC(I+1)*BB(I+2)
IF (IPT .EQ. 0) GO TC 100
PRT(1)=R1
PRT(2)=R2
IF (DABS(PR(I)) .GT. 1.D-10) GO TO 90
PRT(1)=BLANK
PRT(2)=BLANK
PRT(3)=CS
PRT(4)=ED
WRITE(6,18C) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
PRT(3)=SN
I=I+1
WRITE(6,18C) PR(I),PI(I),RES(I),(PRT(J),J=1,4)
PRT(1)=T1
PRT(2)=R2
IF (DABS(PR(I)) .LT. 1.D-10) PRT(2)=BLANK
PRT(3)=CS
I=I+1
WRITE(6,15C) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
PRT(3)=SN
I=I+1
WRITE(6,15C) PR(I),PI(I),RES(I),PRT(1),K,(PRT(J),J=2,4)
GO TO 30
I=I+3
GO TO 30
1CC
C----- COMPUTE REPEATED REAL PCLE RESIDUE AND PRINT CUT ALL K OF THEM-----
110 CONTINUE
KT=I+K-1
NN=0
DO 130 J=I,KT
NN=NN+1
RES(J)=ZERC
CO 120 JJ=J,KT
RES(J)=RES(J)+BB(JJ)*CC(JJ-NN+1)
CCNT INUE
IF (IPT .EQ. 0) GO TC 150
NN=0
PRT(1)=T1
PRT(2)=R2
PRT(3)=BLANK
PRT(4)=BLANK

```



```

DC 140 J=I,KT
WRITE (6,150) PR(J),PI(J),RES(J),PRT(1),NN,(PRT(JJ),JJ=2,4)
NN=NN+1
GO TO 30
150 I=KT
GO TO 30
160 CCNTINUE
RETURN
C-----
170 FORMAT (//,3X,22HRESIDUES AT THE POLES:/,T16,5HF O L E S,T41,15HR
1E S I C U E S,/T9,7HREAL(A),T26,7HIMAG(B))
180 FORMAT (//,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),3A8,A1)
190 FORMAT (//,4X,1H(,F13.6,4H)+J(,F13.6,1H),4X,1H(,F13.6,1H),A4,I2,2X,
12A8,A1)
END
C=====
SUBROUTINE BALANC (NM,N,A,LOW,IGH,SCALE)
INTEGER I,J,K,L,M,N,JJ,NM,IGH,LOW,IEXC
REAL*8 A(NM,N),SCALE(N)
REAL*8 C,F,G,R,S,B2,RADIX
LOGICAL NOCCNV
DATA RADIX/24210000000000000000/
C-----
B2=RADIX*RADIX
K=1
L=N
GO TO 60
IN-LINE PROCEDURE FOR ROW AND COLUMN EXCHANGE-----
1C SCALE(M)=J
IF (J .EQ. N) GO TO 40
DO 20 I=1,L
F=A(I,J)
A(I,J)=A(I,M)
A(I,M)=F
CONTINUE
CC 30 I=K,N
F=A(J,I)
A(J,I)=A(M,I)
A(M,I)=F
CONTINUE
GO TO (50,5C),IEXC
40 SEARCH FOR FOWS ISCLATING AN EIGENVALUE AND PUSH THEM DOWN-----
5C IF (L .EQ. 1) GO TO 230
L=L-1
DO 80 JJ=1,L
J=L+1-JJ
DO 70 I=1,L
OPT222570
OPT222580
OPT222590
OPT222600
OPT222610
OPT222620
OPT222630
OPT222640
OPT222650
OPT222660
OPT222670
OPT222680
OPT222690
OPT222700
OPT222710
OPT222720
OPT222730
OPT222740
OPT222750
OPT222760
OPT222770
OPT222780
OPT222790
OPT222800
OPT222810
OPT222820
OPT222830
OPT222840
OPT222850
OPT222860
OPT222870
OPT222880
OPT222890
OPT222900
OPT222910
OPT222920
OPT222930
OPT222940
OPT222950
OPT222960
OPT222970
OPT222980
OPT222990
OPT23000
OPT230010
OPT230020
OPT230030
OPT230040

```

```

7C      IF (I.EQ. J) GO TO 70
        IF (A(J,I) .NE. 0.000) GO TO 80
        CONTINUE
        M=L
        IEXC=1
        GO TO 10
8C      CCNT INUE
        GO TO 100
9C      C-----SEARCH FOR COLUMNS ISOLATING AN EIGENVALUE AND PUSH THEM LEFT-----
10C      K=K+1
        DO 120 J=K,L
        DO 110 I=K,L
        IF (I.EQ. J) GO TO 110
        IF (A(I,J) .NE. 0.000) GO TO 120
        CCNT INUE
        M=K
        IEXC=2
        GO TO 10
12C      C-----NOW BALANCE THE SUBMATRIX IN ROWS K TO L-----
        DO 130 I=K,L
        SCALE(I)=1.000
13C      C-----ITERATIVE LOOP FOR NORM REDUCTION-----
14C      NCONV=.FALSE.
        DO 220 I=K,L
        C=0.000
        R=0.000
        DO 150 J=K,L
        IF (J.EQ. I) GO TO 150
        C=C+DABS(A(J,I))
        R=R+DABS(A(I,J))
        CCNT INUE
15C      C-----GUARD AGAINST ZERO C OR R DUE TO UNDERFLOW-----
        IF (C.EQ. 0.000 .OR. R.EQ. 0.000) GO TO 220
        G=R/RADIX
        F=1.000
        S=C+R
        IF (C.GE. G) GO TO 170
        F=F*RADIX
        C=C*B2
        GO TO 160
16C      G=R*RADIX
        IF (C.LT. G) GO TO 190
        F=F/RADIX
        C=C/B2
        GO TO 180
17C      G=R*RADIX
        IF (C.LT. G) GO TO 190
        F=F/RADIX
        C=C/B2
        GO TO 180
18C      C-----NOW BALANCE-----
19C      IF ((C + R) / F .GE. 0.9500 * S) GO TO 220

```

```

G=1.0D0/F
SCALE(I)=SCALE(I)*F
NCONV=.TRUE.
DO 200 J=K,N
  A(I,J)=A(I,J)*G
200  A(I,J)=A(I,J)*G
210  A(J,I)=A(J,I)*F
220  CCNTINUE
  IF (NCONV) GO TO 140
  LGW=K
  IGH=L
  RETURN
END
C=====
SUBROUTINE CRTHES (NM,N,LOW,IGH,A,ORT)
  INTEGER I,J,M,N,II,JJ,LA,MP,NM,IGH,KPI,LCW
  REAL*8 A(NM,N),CRT(IGH)
  REAL*8 F,G,F,SCALE
  REAL*8 DSQRT,CABS,DSIGN
  LA=IGH-1
  KPI=LOW+1
  IF (LA .LT. KPI) GO TO 100
  DO 90 M=KPI,LA
    H=0.0D0
    CRT(M)=0.0D0
    SCALE=0.0D0
    DO 10 I=M,IGH
      CC=SCALE+DABS(A(I,M-1))
      IF (SCALE .EQ. 0.0D0) GO TO 90
      MP=M+IGH
      DO 20 II=M,IGH
        I=MP-II
        CRT(I)=A(I,M-1)/SCALE
        H=H+ORT(I)*CRT(I)
        CCNTINUE
        G=-DSIGN(DSQRT(H),CRT(M))
        H=H-ORT(M)*G
        CRT(M)=ORT(M)-G
      DO 50 J=M,N
        F=0.0D0
        DO 30 I=M,IGH
          I=MP-I
          F=F+ORT(I)*A(I,J)
          CCNTINUE
          F=F/H
        DO 40 I=M,IGH

```

122


```

IF (L.EQ. EN) GO TC 220
Y=H(NA,NA)
W=H(EN,NA)*F(NA,EN)
IF (L.EQ. NA) GO TC 230
IF (ITS.EC. 30) GO TO 500
IF (ITS.NE. 10) AND. ITS.NE. 20) GO TO 80
C-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=H(I,I)-X
S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
X=0.75D0*S
Y=X
W=-0.4375D0*S*S
ITS=ITS+1
LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.
DO 90 MM=L,ENM2
N=ENM2+L-MM
Z=H(M,M)
R=X-ZZ
S=Y-ZZ
P=(R*-S-H(M+1,M)+H(M,M+1))
Q=H(M+1,M+1)-ZZ-R-S
R=H(M+2,M+1)
S=DABS(P)+DABS(R)
P=P/S
Q=Q/S
R=R/S
IF (M.EQ. L) GO TO 100
IF (DABS(H(M,M-1))* (DABS(Q) + DABS(R)) .LE. MACHEP * DABS(P)
1 * (DABS(H(M-1,M-1)) + DABS(ZZ) + DABS(H(M+1,M+1)))) GO TO 100
SC
100
CCNT=INUE
MP2=M+2
DO 110 I=MP2,EN
H(I,I-2)=0.D0
IF (I-3).EQ. MP2) GO TC 110
H(I,I-3)=0.D0
CCNT=INUE
C-----DOUBLE CR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
DO 210 K=M,NA
NOTLAS=K.NE.NA
IF (K.EC. N) GO TC 120
P=H(K,K-1)
Q=H(K+1,K-1)
R=0.D0
IF (NOTLAS) R=H(K+2,K-1)
X=DABS(P)+DABS(Q)+DABS(R)
IF (X.EQ. C.DD0) GO TO 210
OPT 24970
OPT 24980
OPT 24990
OPT 25000
OPT 25010
OPT 25020
OPT 25030
OPT 25040
OPT 25050
OPT 25060
OPT 25070
OPT 25080
OPT 25090
OPT 25100
OPT 25110
OPT 25120
OPT 25130
OPT 25140
OPT 25150
OPT 25160
OPT 25170
OPT 25180
OPT 25190
OPT 25200
OPT 25210
OPT 25220
OPT 25230
OPT 25240
OPT 25250
OPT 25260
OPT 25270
OPT 25280
OPT 25290
OPT 25300
OPT 25310
OPT 25320
OPT 25330
OPT 25340
OPT 25350
OPT 25360
OPT 25370
OPT 25380
OPT 25390
OPT 25400
OPT 25410
OPT 25420
OPT 25430
OPT 25440

```

```

P=P/X
Q=Q/X
R=R/X
S=DSIGN(DSGRT(P*P+Q*Q+R*R),P)
120 IF (K.EQ. N) GC TC 130
H(K,K-1)=-S*X
GC TO 140
130 IF (L.NE. N) H(K,K-1)=-H(K,K-1)
P=P+S
X=P/S
Y=Q/S
ZZ=R/S
Q=Q/P
R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,N
P=H(K,J)+Q*H(K+1,J)
IF (.NOT. NCTLAS) GO TC 150
P=P+R*H(K+2,J)
H(K+2,J)=H(K+2,J)-P*ZZ
H(K+1,J)=H(K+1,J)-P*Y
H(K,J)=H(K,J)-P*X
150 CCNT INUE
160 J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
DO 180 I=1,J
P=X*H(I,K)+Y*H(I,K+1)
IF (.NOT. NCTLAS) GO TO 170
P=P+ZZ*H(I,K+2)
H(I,K+2)=H(I,K+2)-P*R
H(I,K+1)=H(I,K+1)-P*C
H(I,K)=H(I,K)-P
170 CCNT INUE
180 C-----ACCUMULATE TRANSFORMATIONS-----
DO 200 I=LCW,IGH
P=X*Z(I,K)+Y*Z(I,K+1)
IF (.NOT. NCTLAS) GC TO 190
P=P+ZZ*Z(I,K+2)
Z(I,K+2)=Z(I,K+2)-P*R
Z(I,K+1)=Z(I,K+1)-P*C
Z(I,K)=Z(I,K)-P
190 CCNT INUE
200 GO TO 40
210 C-----ONE ROOT FOUND-----
H(EN,EN)=X+Y
WR(EN)=H(EN,EN)
W(EN)=C.ODC
220

```

```

EN=NA
GO TO 30
C-----TWO ROOTS FOUND-----
230 P=(Y - X)/2.0D0
C=P*P+W
ZZ=DSQRT(DABS(Q))
H(EN,EN)=X+T
X=H(EN,EN)
H(NA,NA)=Y+T
IF (Q .LT. 0.0D0) GO TO 270
C-----REAL PAIR-----
ZZ=P+DSIGN(ZZ,P)
WR(NA)=X+ZZ
WR(EN)=WR(NA)
IF (ZZ .NE. 0.0D0) WR(EN)=X-W/ZZ
WI(NA)=0.0D0
WI(EN)=0.0D0
X=H(EN,NA)
S=DABS(X)+LABS(ZZ)
P=X/S
Q=ZZ/S
R=DSQRT(P*P+Q*Q)
P=P/R
C=Q/R
C-----ROW MODIFICATION-----
DO 240 J=NA,N
ZZ=H(NA,J)
H(NA,J)=Q*ZZ+P*H(EN,J)
H(EN,J)=Q*P(EN,J)-P*ZZ
CONTINUE
240 C-----COLUMN MODIFICATION-----
DO 250 I=1,EN
ZZ=H(I,NA)
H(I,NA)=Q*ZZ+P*H(I,EN)
H(I,EN)=Q*P(I,EN)-P*ZZ
CONTINUE
250 C-----ACCUMULATE TRANSFORMATIONS-----
DO 260 I=LCH,IGH
ZZ=Z(I,NA)
Z(I,NA)=Q*ZZ+P*Z(I,EN)
Z(I,EN)=Q*Z(I,EN)-P*ZZ
CCN+INUE
GO TO 280
260 C-----COMPLEX PAIR-----
270 WR(NA)=X+P
WR(EN)=X+P
WI(NA)=ZZ
WI(EN)=-ZZ

```

```

280 EN=ENM2
C GO TO 30
C-----ALL ROOTS FOUND. BACKSUBSTITUTE TO FIND-----
C-----VECTORS OF UPPER TRIANGULAR FORM-----
290 IF (NORM.EQ.C.ODD) GO TO 510
C-----
C-----REAL VECTOR-----
300 N=EN
H(EN,EN)=1.CDD
IF (NA.EQ.O) GO TO 450
DO 360 II=1,NA
I=EN-II
W=H(I,I)-P
R=H(I,EN)
IF (M.GT.NA) GO TO 320
DO 310 J=M,NA
R=R+H(I,J)*F(J,EN)
IF (WI(I).GE.O.ODD) GO TO 330
ZZ=W
S=R
GO TO 360
230 M=I
IF (WI(I).NE.O.ODD) GO TO 340
T=W
IF (W.EQ.C.ODD) T=MACHEP*NORM
H(I,EN)=-R/T
GO TO 360
C-----SOLVE REAL EQUATIONS-----
240 X=H(I,I+1)
Y=H(I+1,I)
T=(WR(I)-P)+WI(I)*WI(I)
C=(X*S-ZZ*R)/Q
H(I,EN)=T
IF (DABS(X).LE.DABS(ZZ)) GO TO 350
H(I+1,EN)=(-R-W*T)/X
GO TO 360
250 H(I+1,EN)=(-S-Y*T)/ZZ
C CONTINUE
C-----END REAL VECTOR-----
C-----COMPLEX VECTOR-----
270 M=NA
C-----LAST VECTOR COMPONENT CHOSEN IMAGINARY SC THAT-----
OPT26410
OPT26420
OPT26430
OPT26440
OPT26450
OPT26460
OPT26470
OPT26480
OPT26490
OPT26500
OPT26510
OPT26520
OPT26530
OPT26540
OPT26550
OPT26560
OPT26570
OPT26580
OPT26590
OPT26600
OPT26610
OPT26620
OPT26630
OPT26640
OPT26650
OPT26660
OPT26670
OPT26680
OPT26690
OPT26700
OPT26710
OPT26720
OPT26730
OPT26740
OPT26750
OPT26760
OPT26770
OPT26780
OPT26790
OPT26800
OPT26810
OPT26820
OPT26830
OPT26840
OPT26850
OPT26860
OPT26870
OPT26880

```



```

C-----EIGENVECTR MATRIX IS TRIANGULAR-----
IF (DABS(H(EN,NA)) .LE. DABS(H(NA,EN))) GC TC 380
H(NA,NA)=Q/F(EN,NA)
H(NA,EN)=-(F(EN,EN) - P)/H(EN,NA)
GO TO 390
380 Z3=DCMLPX(C.ODO,-H(NA,EN))/DCMLPX(H(NA,NA)-P,Q)
H(NA,NA)=DREAL(Z3)
H(NA,EN)=DIMAG(Z3)
H(EN,NA)=C.ODO
H(EN,EN)=1.ODO
ENM2=NA-1
IF (ENM2 .EQ. 0) GC TO 450
CO 440 II=I,ENM2
I=NA-I
W=H(I,I)-P
RA=O.ODO
SA=H(I,EN)
DO 400 J=M,NA
RA=RA+H(I,J)*H(J,NA)
SA=SA+H(I,J)*H(J,EN)
CCNT INUE
IF (WI(I) .GE. O.ODO) GO TO 410
ZZ=W
ZR=RA
Z=SA
GO TO 440
410 M=I
IF (WI(I) .NE. O.ODO) GO TO 420
Z3=DCMLPX(-RA,-SA)/DCMLPX(W,Q)
H(I,NA)=DREAL(Z3)
H(I,EN)=DIMAG(Z3)
GO TO 440
C-----SOLVE COMPLEX EQUATIONS-----
420 X=H(I,I+1)
Y=H(I+1,I)
VR=(WR(I) - P)*WI(I)*WI(I)-Q*Q
VI=(WR(I) - P)*2.ODO*Q
IF (VR .EQ. O.ODO .AND. VI .EQ. O.ODO) VR=MACFEP*NORM*(DABS(W) + D
1ABS(Q) + DABS(X) + DABS(Y) + DABS(ZZ))
Z3=DCMLPX(X*R-ZZ*RA+C*SA,X*S-ZZ*SA-Q*RA)/DCMLPX(VR,VI)
H(I,NA)=DREAL(Z3)
H(I,EN)=DIMAG(Z3)
P=(I,EN)=DREAL(Z3)
IF (DABS(X) .LE. DABS(ZZ) + DABS(Q)) GO TO 430
H(I+1,NA)=(-RA - W * H(I,NA) + Q * H(I,EN))/X
H(I+1,EN)=(-SA - W * H(I,EN) - Q * H(I,NA))/X
GO TO 440
430 Z3=DCMLPX(-R-Y*H(I,NA),-S-Y*H(I,EN))/DCMLPX(ZZ,C)
H(I+1,NA)=CFEAL(Z3)

```



```

440 H(I+1,EN)=CIMAG(Z3)
C CONTINUE-----END COMPLEX VECTOR-----OPT27370
450 CONTINUE-----ENC BACK SUBSTITUTION. VECTORS OF ISCLATED ROOTS-----OPT27380
C-----OPT27390
DO 470 I=1,N
IF (I .GE. LOW .AND. I .LE. IGH) GO TO 470
DC 460 J=I,N
Z(I,J)=H(I,J)
C CONTINUE-----MULTIPLY BY TRANSFORMATION MATRIX TO GIVE-----OPT27400
C-----VECTORS OF ORIGINAL FULL MATRIX.-----OPT27410
DO 490 JJ=LCW,N
J=N+LOW-JJ
M=MINO(J,I,GH)
DO 490 I=LCW,IGH
ZZ=0.000
DO 480 K=LCW,M
ZZ=ZZ+Z(I,K)*H(K,J)
Z(I,J)=ZZ
C CONTINUE-----OPT27420
GO TO 510
C-----SET ERROR --->NO CONVERGENCE TO AN-----OPT27430
C-----EIGENVALUE AFTER 30 ITERATIONS-----OPT27440
IFERR=EN
RETURN
C-----OPT27450
C-----OPT27460
C-----OPT27470
C-----OPT27480
C-----OPT27490
C-----OPT27500
C-----OPT27510
C-----OPT27520
C-----OPT27530
C-----OPT27540
C-----OPT27550
C-----OPT27560
C-----OPT27570
C-----OPT27580
C-----OPT27590
C-----OPT27600
C-----OPT27610
C-----OPT27620
C-----OPT27630
C-----OPT27640
C-----OPT27650
C-----OPT27660
C-----OPT27670
C-----OPT27680
C-----OPT27690
C-----OPT27700
C-----OPT27710
C-----OPT27720
C-----OPT27730
C-----OPT27740
C-----OPT27750
C-----OPT27760
C-----OPT27770
C-----OPT27780
C-----OPT27790
C-----OPT27800
C-----OPT27810
C-----OPT27820
C-----OPT27830
C-----OPT27840

```

```

S=Z(I,J)
Z(I,J)=Z(K,J)
Z(K,J)=S
4C CCNTINUE
5C RETURN
6C END
C=====
SUBROUTINE FQR (NM,N,LOW,IGH,H,WR,WI,IERR)
  INTEGER I,J,K,L,M,N,EN,LL,MM,NA,NM,IGH,ITS,LCW,MP2,ENM2,IERR
  REAL*8 H(NM,N),WR(N),WI(N)
  REAL*8 P,Q,R,S,T,W,X,Y,ZZ,NORM,MACHEP
  REAL*8 DSQRT,DABS,DSIGN
  INTEGER MINO
  LOGICAL NOTLAS
  DATA MACHEP/Z3410000000000000/
  IERR=0
  NORM=0.0D0
  K=1
C----- STORE RCCTS ISOLATED BY BALANC AND COMPUTE MATRIX NORM-----
DO 20 I=1,N
DO 10 J=K,N
  NORM=NORM+DABS(H(I,J))
10 K=I
  IF (I .GE. LOW .AND. I .LE. IGH) GO TO 20
  WR(I)=H(I,I)
  WI(I)=0.0D0
  CCNTINUE
  EN=IGH
  T=0.0D0
C----- SEARCH FOR NEXT EIGENVALUES-----
3C IF (EN .LT. LOW) GO TO 250
  ITS=0
  NA=EN-1
  ENM2=NA-1
C----- LOCK FOR SINGLE SMALL SUR-DIAGCNAL ELEMENT-----
4C DO 50 LL=LCW,EN
  L=EN+LCW-LL
  IF (L .EQ. LOW) GO TO 60
  S=DABS(H(L-1,L-1))+DABS(H(L,L))
  IF (S .EQ. 0.0D0) S=NORM
  IF (DABS(H(L,L-1)) .LE. MACHEP * S) GO TO 60
  CCNTINUE
50
C----- FORM SHIFT-----
6C X=H(EN,EN)
  IF (L .EQ. EN) GO TO 200
  Y=H(NA,NA)
  W=H(EN,NA)*F(NA,EN)

```

OPT27850
 OPT27860
 OPT27870
 OPT27880
 OPT27890
 OPT27900
 OPT27910
 OPT27920
 OPT27930
 OPT27940
 OPT27950
 OPT27960
 OPT27970
 OPT27980
 OPT27990
 OPT28000
 OPT28010
 OPT28020
 OPT28030
 OPT28040
 OPT28050
 OPT28060
 OPT28070
 OPT28080
 OPT28090
 OPT28100
 OPT28110
 OPT28120
 OPT28130
 OPT28140
 OPT28150
 OPT28160
 OPT28170
 OPT28180
 OPT28190
 OPT28200
 OPT28210
 OPT28220
 OPT28230
 OPT28240
 OPT28250
 OPT28260
 OPT28270
 OPT28280
 OPT28290
 OPT28300
 OPT28310
 OPT28320

```

IF (L.EQ.NA) GO TO 210
IF (ITS.EQ.30) GO TO 240
IF (ITS.NE.10.AND.ITS.NE.20) GO TO 80
-----FORM EXCEPTIONAL SHIFT-----
T=T+X
DO 70 I=LOW,EN
H(I,I)=H(I,I)-X
S=DABS(H(EN,NA))+DABS(H(NA,ENM2))
X=0.75D0*S
Y=X
W=-0.4375D0*S*S
ITS=ITS+1
-----LOOK FOR TWO CONSECUTIVE SMALL SUB-DIAGONAL ELEMENTS.-----
DO 90 MM=L,ENM2
N=ENM2+L-MM
ZZ=H(M,M)
R=X-ZZ
S=Y-ZZ
P=(R*S-W)/H(M+1,M)+H(M,M+1)
Q=H(M+1,M+1)-ZZ-R-S
R=H(M+2,M+1)
S=DABS(P)+DABS(Q)+DABS(R)
P=P/S
Q=Q/S
R=R/S
IF (M.EQ.L) GO TO 100
IF (DABS(P(M,M-1))*DABS(Q)+DABS(R))*LE.MACHEP*DABS(P)
1 * (DABS(H(M-1,M-1))+DABS(ZZ)+DABS(H(M+1,M+1))) GO TO 100
9C
100 CONTINUE
MP2=M+2
DO 110 I=MP2,EN
H(I,I-2)=0.D0
IF (I.EQ.MP2) GO TO 110
H(I,I-2)=0.D0
110 CONTINUE
-----DOUBLE QR STEP INVOLVING ROWS L TO EN AND COLUMNS M TO EN-----
DO 190 K=M,NA
NCTLAS=K.NE.NA
IF (K.EQ.M) GO TO 120
P=H(K,K-1)
Q=H(K+1,K-1)
R=0.D0
IF (NCTLAS) R=H(K+2,K-1)
X=DABS(P)+DABS(Q)+DABS(R)
IF (X.EQ.0.D0) GO TO 190
P=P/X
Q=Q/X
R=R/X

```

```

120 S=DSIGN(DSQRT(P*P+Q*(C+R*R),P)
    IF (K.EQ.0) GO TO 130
    H(K,K-1)=-S*X
    GO TO 140
120 IF (L.NE.0) H(K,K-1)=-H(K,K-1)
140 P=P+S
    X=P/S
    Y=Q/S
    ZZ=R/S
    Q=Q/P
    R=R/P
C-----ROW MODIFICATION-----
DO 160 J=K,EN
    P=H(K,J)+Q*(K+1,J)
    IF (.NOT. NCTLAS) GO TO 150
    P=P+R*(K+2,J)
    H(K+2,J)=H(K+2,J)-P*ZZ
    H(K+1,J)=H(K+1,J)-P*Y
    H(K,J)=H(K,J)-P*X
    CONTINUE
    J=MINO(EN,K+3)
C-----COLUMN MODIFICATION-----
CC 180 I=L,J
    P=X*(H(I,K)+Y*(H(I,K+1)
    IF (.NOT. NCTLAS) GO TO 170
    P=P+ZZ*(H(I,K+2)
    H(I,K+2)=H(I,K+2)-P*R
    H(I,K+1)=H(I,K+1)-P*Q
    H(I,K)=H(I,K)-P
    CONTINUE
    CONTINUE
    GO TO 40
C-----ONE ROOT FOUND-----
200 WR(EN)=X+T
    WI(EN)=0.0DC
    EN=NA
    GO TO 30
C-----TWO ROOTS FOUND-----
210 P=(Y-X)/2.0D0
    Q=P*(P+W)
    ZZ=DSQRT(DABS(Q))
    X=X+T
    IF (Q.LT.0.0D0) GO TO 220
C-----REAL PAIR-----
    ZZ=P+DSIGN(ZZ,P)
    WR(NA)=X+ZZ
    WR(EN)=WR(NA)
    IF (ZZ.NE.0.0D0) WR(EN)=X-W/ZZ

```

OPT28810
OPT28820
OPT28830
OPT28840
OPT28850
OPT28860
OPT28870
OPT28880
OPT28890
OPT28900
OPT28910
OPT28920
OPT28930
OPT28940
OPT28950
OPT28960
OPT28970
OPT28980
OPT28990
OPT29000
OPT29010
OPT29020
OPT29030
OPT29040
OPT29050
OPT29060
OPT29070
OPT29080
OPT29090
OPT29100
OPT29110
OPT29120
OPT29130
OPT29140
OPT29150
OPT29160
OPT29170
OPT29180
OPT29190
OPT29200
OPT29210
OPT29220
OPT29230
OPT29240
OPT29250
OPT29260
OPT29270
OPT29280


```

C-----COMPLEX PAIR-----
220  WI(NA)=0.0D0
    WI(EN)=0.0DC
    GO TO 230
    OPT292900
    OPT29310
    OPT29330
    OPT29330
    OPT29340
    OPT29350
    OPT29360
    OPT29370
    OPT29380
    OPT29390
    OPT29400
    OPT29410
    OPT29420
    OPT29430
    OPT29440
    OPT29450
    OPT29460
    OPT29470
    OPT29480
    OPT29490
    OPT29500
    OPT29510
    OPT29520
    OPT29530
    OPT29540
    OPT29550
    OPT29560
    OPT29570
    OPT29580
    OPT29590
    OPT29600
    OPT29610
    OPT29620
    OPT29630
    OPT29640
    OPT29650
    OPT29660
    OPT29670
    OPT29680
    OPT29690
    OPT29700
    OPT29710
    OPT29720
    OPT29730
    OPT29740
    OPT29750
    OPT29760

C-----COMPLEX PAIR-----
220  WR(NA)=X+P
    WR(EN)=X+P
    WI(NA)=ZZ
    WI(EN)=ZZ
    EN=ENM2
    GO TO 30
    OPT292900
    OPT29310
    OPT29330
    OPT29330
    OPT29340
    OPT29350
    OPT29360
    OPT29370
    OPT29380
    OPT29390
    OPT29400
    OPT29410
    OPT29420
    OPT29430
    OPT29440
    OPT29450
    OPT29460
    OPT29470
    OPT29480
    OPT29490
    OPT29500
    OPT29510
    OPT29520
    OPT29530
    OPT29540
    OPT29550
    OPT29560
    OPT29570
    OPT29580
    OPT29590
    OPT29600
    OPT29610
    OPT29620
    OPT29630
    OPT29640
    OPT29650
    OPT29660
    OPT29670
    OPT29680
    OPT29690
    OPT29700
    OPT29710
    OPT29720
    OPT29730
    OPT29740
    OPT29750
    OPT29760

C-----SET ERROR -- NO CONVERGENCE TO AN
C-----EIGENVALUE AFTER 30 ITERATIONS-----
240  IERR=EN
250  RETURN
    OPT292900
    OPT29310
    OPT29330
    OPT29330
    OPT29340
    OPT29350
    OPT29360
    OPT29370
    OPT29380
    OPT29390
    OPT29400
    OPT29410
    OPT29420
    OPT29430
    OPT29440
    OPT29450
    OPT29460
    OPT29470
    OPT29480
    OPT29490
    OPT29500
    OPT29510
    OPT29520
    OPT29530
    OPT29540
    OPT29550
    OPT29560
    OPT29570
    OPT29580
    OPT29590
    OPT29600
    OPT29610
    OPT29620
    OPT29630
    OPT29640
    OPT29650
    OPT29660
    OPT29670
    OPT29680
    OPT29690
    OPT29700
    OPT29710
    OPT29720
    OPT29730
    OPT29740
    OPT29750
    OPT29760

C=====
1  SUBROUTINE PSDCAL (N2,NS,FA,X,NC,GW,GV,C,NC,HY,HU,H,
2  FBGE,NG,GAM,ACL,F,WR,WI,DI,D2,JCF,RES,Q,R,EP,CC,IYU,
   IPSD,INORM)
   =====
   PSDCAL COMPUTES THE PSD OF OUTPUTS CR CONTROLS OF
   A CONTROLLED SYSTEM
   =====
   IYU= 1  CUTPUT PSD
       2  CONTROL PSD
       3  BOTH OUTPUT AND CONTROL PSD
   IPSD=1  PSD AND TF RESIDUES
       2  PSD AND TF RESIDUES
   INORM= 1,2,0.00 NG NORMALIZED BY ITH PROCESS NOISE
        NG+1,0.00 NG+NO NORMALIZED BY ITH MEAS NOISE
   =====
1  DOUBLE PRECISION FA,X,GW,GV,C,HY,H,FBGE,GAM,ACL,F,WR,WI,DI,D2,RES,
1BB,CC,G,R,PSD,W,DNORM,DNI,EMAX,ELOG,EMOD,DW,ST,CM,RE,AI,HU,DW1
C  COMPLEX *16 ZN,ZZ
C  DIMENSION FA(N2,N2),X(N2,N2),GW(N2,NG),C(NC,NS),HY(NC,NS),H(NC,NS),
1  FBGE(NS,NC),GAM(NS,NG),ACL(NS,NS),F(NS,NS),WR(N2),WI(N2),DOPT29660
22(N2),RES(N2),Q(NG,NG),R(NC,NC),PSD(30),W(30),BE(N2),CC(N2),GV(N2),
3  NO),HU(NC,N2),DW1(4)
C  INTEGER JCF(N2)
C  DATA DW1/1.0D0,5.0D0,10.0D0/
   IF (IYU.EQ.0) IYU=1
   IF (INORM.EQ.0) INORM=1
   IPT=0
   IF (IPSD.GT.1) IPT=1
   IX=INORM-NG
   =====
    OPT292900
    OPT29310
    OPT29330
    OPT29330
    OPT29340
    OPT29350
    OPT29360
    OPT29370
    OPT29380
    OPT29390
    OPT29400
    OPT29410
    OPT29420
    OPT29430
    OPT29440
    OPT29450
    OPT29460
    OPT29470
    OPT29480
    OPT29490
    OPT29500
    OPT29510
    OPT29520
    OPT29530
    OPT29540
    OPT29550
    OPT29560
    OPT29570
    OPT29580
    OPT29590
    OPT29600
    OPT29610
    OPT29620
    OPT29630
    OPT29640
    OPT29650
    OPT29660
    OPT29670
    OPT29680
    OPT29690
    OPT29700
    OPT29710
    OPT29720
    OPT29730
    OPT29740
    OPT29750
    OPT29760

```



```

IF (IX .GT. 0) WRITE (6,330) IX
IF (IX .LE. 0) WRITE (6,340) INORM
NSQ=N2*N2
C----- COMPUTE EIGENSYSTEM OF CONTROLLED SYSTEM; FCRM FA-----
DO 10 I=1,N2
DO 10 J=1,N2
FA(I,J)=ACL(I,J)
FA(NS+I,J)=C.DD
DO 30 I=1,N2
DO 30 J=1,N2
ST=0.DD
DO 20 K=1,N2
ST=ST+FBGE(I,K)*H(K,J)
FA(I,NS+J)=-ST
FA(NS+I,NS+J)=F(I,J)-ST
CALL RAPRNT (N2,N2,N2,9,FA,4,'(9(IX,1PD13.6))')
C-----
CALL BALANC (N2,N2,FA,LOW,IHIGH,D1)
CALL ORTHES (N2,N2,LCW,IHIGH,FA,D2)
CALL ORTRAN (N2,N2,LCW,IHIGH,FA,D2,X)
CALL HQR2 (N2,N2,LOW,IHIGH,FA,WR,WI,X,IERR)
IF (IERR .NE. 0) GO TO 320
CALL BALBAK (N2,N2,LCW,IHIGH,D1,N2,X)
CALL RAPRNT (N2,N2,N2,5,X,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE; DETERMINE MODAL MATRICES-----
IF (IYU .EQ. 1) GO TO 60
DO 50 I=1,N2
DO 50 J=1,N2
ST=0.DD
DO 40 K=1,N2
ST=ST-C(I,K)*X(K,J)
HU(I,J)=ST
GO TO 90
C----- HSUBU-----
DO 80 I=1,N2
DO 80 J=1,N2
ST=0.DD
DO 70 K=1,N2
ST=ST+H(I,K)*X(K,J)-H(I,K)*X(NS+K,J)
HY(I,J)=ST
CALL RAPRNT (N2,N2,N2,9,HY,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE-----
CALL MINV (NSQ,X,N2,ST,D1,D2)
CALL RAPRNT (N2,N2,N2,5,X,4,'(9(IX,1PD13.6))')
C----- DEBUG ABOVE-----
DC 110 I=1,N2
C----- GSUBW-----

```

```

100 DC 110 J=1,NG
110 ST=0.000
    DO 100 K=1,NS
    ST=ST-X(I,N5+K)*GAM(K,J)
    CALL RPRNT (N2,N2,NG,9,GW,4,'(9(IX,1PDI3.6))')
C-----DEBUG ABOVE: USE SELECTED NORMALIZATION-----
    IF (INORM .LE. NG) DNORM=1.00/Q(INORM,INORM)
    IF (INORM .GT. NG) DNORM=1.00/R(INORM-NG,INORM-NG)
C-----DETERMINE BANDWIDTH OF CONTROLLED SYSTEM-----
    EMAX=0.00
    DO 120 I=1,N2
    EMOD=DABS(WF(I)**2 +WI(I)**2)
    IF (EMOD .GT. EMAX) EMAX=EMOD
    CCNT INUE
    EMOD=DSQRT(EMAX)
    EMOD=2*EMOD
C-----ROUND UP TO NEAREST 2,4,5,8,10-----
    ELOG=DLOG10(EMOD)
    IF (ELOG .LT. 0.00) IPOW=-IDINT(DABS(ELCG) + 1)
    IF (ELOG .GE. 0.00) IPOW=IDINT(ELOG)
    EMAX=EMOD*10**(-IPOW)
    IF (EMAX .GT. 2.00) EMOD=2.00
    IF (EMAX .GT. 4.00) EMOD=4.00
    IF (EMAX .GT. 5.00) EMOD=5.00
    IF (EMAX .GT. 8.00) EMOD=8.00
    IF (EMAX .GE. 10.00) EMOD=10.00
    EMAX=EMOD*10**IPOW
    CW=EMAX/20.00
C-----ADD 10 POINTS 3 DECADES UP-----
    IF (EMOD .LT. 5.0) GC TO 130
    EMAX=1.001
    IK=3
    GC TO 140
    EMAX=5.00
    IK=2
    CCNT INUE
C-----STORE 30 FREQUENCIES-----
    DO 150 I=1,20
    W(I)=DW*(I-1)
    DO 160 J=1,3
    IP=20+2*(I-1)
    DO 160 J=1,3
    IX=MOD(IK+J-1,3)+1
    JJ=0
    IF (IK .EQ. 2 .AND. J .GE. 2) JJ=1
    W(IP+JJ)=DWI(IX)*10**((IPOW+I-1+JJ+IK-2)
    CCNT INUE
160

```

```

IX=MOD(IK,2)+1
W(30)=DW1(I)*10*(IPOW+3 +IK-2)
-----LARGE LOOP THRU OUTPUTS-----
IF (IYU.EQ.1) NL=NC
IF (IYU.EQ.2) NL=NC
DO 310 L=1,NL
DO 170 I=1,20
PSD(I)=0.0
-----LCCP THRU PROCESS NOISE-----
DO 220 I=1,NG
CN1=DNCRM*(I,I)
IF (IYU.EQ.1) AND. IPT .EQ. 1) WRITE (6,350) I,L
IF (IYU.EQ.2) AND. IPT .EQ. 1) WRITE (6,360) I,L
IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GW,NL,HY,WR,WI,
1RES,BB,CC,IFT)
IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NG,GW,NL,HL,WR,WI,
1RES,BB,CC,IFT)
DO 210 K=1,20
ZZ=DCMPLX(0.0,0.0)
OM=W(K)
DO 200 II=1,N2
IF (WI(II)) 200,180,190
ZZ=DCMPLX(-WR(II),OM-WI(II))
ZZ=RES(II)/ZZ+ZZ
GO TO 200
RE=WR(II)
AI=WI(II)
ZN=DCMPLX(RES**2 + AI**2 - OM**2,-2.0*RE*OM)
ZZ=ZN+ZN/ZI
CCNT=INUE
PSD(K)=PSD(K)+DN1*(ZZ*DCONJG(ZZ))
CCNT=INUE
-----G SUBV-----
DO 240 I=1,N2
DO 240 J=1,NC
ST=0.0
DO 230 K=1,NS
ST=ST+X(I,K)*FBGE(K,J)+X(I,NS+K)*FBGE(K,J)
GV(I,J)=ST
CALL RAPRNT (N2,N2,NC,9,GV,4,'(9(IX,1PD13.6))')
DERUG ABCVE, LOOP THRU MEAS NOISE
DO 300 I=1,NC
CN1=DNCRM*(I,I)
IF (IYU.EQ.1) AND. IPT .EQ. 1) WRITE (6,370) I,L
IF (IYU.EQ.2) AND. IPT .EQ. 1) WRITE (6,380) I,L
IF (IYU.EQ.1) CALL RESID (I,L,N2,JCF,NG,GV,NL,HY,WR,WI,RES,
1BB,CC,IFT)
1

```

```

1 IF (IYU.EQ.2) CALL RESID (I,L,N2,JCF,NO,GV,NL,HU,WR,WI,RES,
  DO 290 K=1,20
  ZZ=DCMPLX(C.DC,0.DC)
  CM=W(K)
  DO 270 II=1,N2
  IF (WI(II)) 27C,25C,26C
  ZD=DCMPLX(-WR(II),CM-WI(II))
  ZZ=ZZ+RES(II)/ZD
  GO TO 270
  RE=WR(II)
  AI=WI(II)
  ZC=DCMPLX(RES**2 + AI**2 -OM**2,-2.DC*RE*OM)
  ZN=DCMPLX(RES(II+1)*AI-RES(II)*RE,RES(II)*CM)
  ZZ=ZZ+ZN/ZC
  CCNTINUE
  IF (IYU.EC.2.OR. I.NE. L) GO TO 280
  PSD(K)=PSD(K)+DN1
  PSD(K)=PSD(K)+DN1*(ZZ#DCONJG(ZZ))
  CCNTINUE
  IF (IYU.EC.1) WRITE (6,390) L
  IF (IYU.EC.2) WRITE (6,400) L
  WRITE (6,410) (W(I),PSD(I),I=1,30)
  CCNTINUE
  RETURN
  CALL EXEXIT (N2,FA,IERR)
  RETURN
-----
33C FORMAT (/,4H SUBSEQUENT PSD IS NORMALIZED BY MEAS NO.,I3,/)
34C FORMAT (/,50H SUBSEQUENT PSD IS NORMALIZED BY PROCESS NCISE NC.,I3,/)
35C J,/)
36C 1ASUREMENT ,I2,/)
37C 1FORMAT (/,36H TRANSFER FUNCTION FROM PROCESS NCISE ,I2,3H TO,13H ME
38C 1TROL ,I2,/)
39C 1MENT ,I2,/)
40C 1FORMAT (/,36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,16H TO MEASURE
41C 1FORMAT (/,36H TRANSFER FUNCTION FROM MEASUREMENT ,I2,12H TO CCNTPOL
42C 1,I2,/)
43C 1FORMAT (/,14H PSD OF OUTPUT,I3,32H FORCED BY ALL NCISE-(RAD FREQ.,
44C 115HNORMALIZED PSD)/)
45C 1FORMAT (/,14H PSD OF CCNTROL,I3,32H FORCED BY ALL NCISE-(RAD FREQ,
46C 1,15HNORMALIZED PSD)/)
47C 1FORMAT (4(I,1H,1H,511.4,1H),)
48C END
=====
C=====

```



```

SUBROUTINE EREXIT (N,A,IERR)
EREXIT RETURNS THE NUMBER OF THE EIGENVALUE WHERE HQR2
FAILS, THEN STOPS THE PROGRAM.
=====
INTEGER IFRF
DOUBLE PRECISION A
DIMENSION A(N,N)
WRITE (5,10) IERR
CALL RAPRNT (N,N,N,9,A,4, '(9(1X,1PD13.6))')
RETURN
FORMAT (35H FAILURE IN HQR2 ON EIGENVALUE NO. ,I3)
END
=====
C=====
SURROUTINE READF (NS,ISAF,BA)
INTERACTIVELY INPUTS THE "F" MATRIX ELEMENT BY ELEMENT.
=====
REAL*8 BA(NS,NS),DUM,ANSR
INTEGER I,J,K,L,IAN,ISAF
DATA IFRF,Y,J,K,L,IAN,ISAF
IF (ISAF.EQ.1) GO TO 40
WRITE (5,120)
DO 20 I=1,NS
DO 10 J=1,NS
WRITE (5,120) I,J
CALL RDCREAL (ANSR)
BA(I,J)=ANSR
CCNT=CCNT+1
GO TO 10
10 CCNT=CCNT+1
20 CCNT=CCNT+1
30 CALL FRTCMS ('CLRSCRN ')
CCNT=CCNT+1
40 WRITE (5,140)
CALL MATPRT (BA,NS,NS)
50 WRITE (5,150)
CALL RDCCHAR (IAN)
IF ((IAN.NE.IY).AND.(IAN.NE.IZ)) GO TO 60
60 GO TO 70
WRITE (5,160)
70 CCNT=CCNT+1
IF ((IAN.NE.IZ) GO TO 110
IF (IAN.NE.IY) GO TO 80
80 WRITE (5,170)
CALL RDCINT (IAN)
K=IAN
WRITE (5,180)
CALL RDCINT (IAN)
L=IAN

```



```

120 WRITE (5,12C) K,L
130 CALL RCREAL (ANSR)
140 DUM=ANSR
150 DC 100 I=1,NS
160 DO 90 J=1,NS
170 IF ((I.EQ.K).AND.(J.EQ.L)) BA(I,J)=DUM
180 CCNT INUE
190 GO TO 30
200 CCNT INUE
210 CALL FRTCMS ('CLRSCRN ')
220 RETURN
230
240 FORMAT (5X,14H THE ELEMENT F(I2,1H,I2,2H)=)
250 FORMAT (//,5X,36H ENTER THE SYSTEM MATRIX "F"-MATRIX,/,10X,41HDIMENSION = # STATES NS& X # STATES NS&)
260 IENSION = # STATES NS& X # STATES NS&
270 FORMAT (//,15X,33H THE SYSTEM MATRIX "F"-MATRIX&...//)
280 FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT? ,//,10X,15H TYPE "YES" OR "NO".)
290 IENT? ,//,10X,15H TYPE "YES" OR "NO".)
300 FORMAT (//,1X,41H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
310 FORMAT (5X,54H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
320 FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED.)
330 I.)
340 END
350
360 SUBROUTINE READH (NO,NS,ISAH,HO)
370 INTERACTIVELY INPUTS THE "H" MATRIX MEASUREMENT SCALING MATRIX&.=
380
390 REAL*8 HO(NO,NS),DUM,ANSR
400 INTEGER IANS,I,J,K,L,ISAH
410 DATA IV,Y,Z,N/
420
430 IF (ISAH.EQ.1) GO TO 40
440 WRITE (5,12C)
450 DC 20 I=1,NS
460 DO 10 J=1,NS
470 WRITE (5,11C) I,J
480 CALL RCREAL (ANSR)
490 HO(I,J)=ANSR
500 CCNT INUE
510 CCNT INUE
520
530 CALL FRTCMS ('CLRSCRN ')
540 CCNT INUE
550 WRITE (5,12C)
560 CALL MATPRT (HO,NO,NS)
570 WRITE (5,14C)
580 CALL RDCHAR (IANS)
590

```

```

6C      IF (( IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
70      GO TO 70
      WRITE (5,15C)
      GO TO 50
      CCNT INUE
      IF (IANS.EC.IZ) GO TC 100
      WRITE (5,16C)
      CALL RCINT (IANS)
      K=IANS
      WRITE (5,17C)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,11C) K,L
      CALL RCREAL (ANSR)
      DUM=ANSR
      DO 90 I=1,NC
      DO 80 J=1,NS
      IF ((I.EQ.K).AND.(J.EQ.L)) HO(I,J)=DUM
8C      CCNT INUE
9C      GO TO 30
100     CCNT INUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
-----
110     FORMAT (5X,14HTHE ELEMENT H(I,I2,1H,I2,2H)=)
120     FORMAT (//,5X,50HENTER THE MEASUREMENT SCALING MATRIX "H"-MATRIX
130     1,//,10X,47HCIMENSION= # OBSERVATIONS NC&X # STATES NS&)
140     FORMAT (//,10X,46HTHE MEASUREMENT SCALING MATRIX "H"-MATRIX&ooo, /
150     1//)
160     FORMAT (//,5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT
170     1ENT?,//,10X,19HTYPE "YES" OR "NO".)
      FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
      FORMAT (5X,50HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
      FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1)
      END
C=====
C      SUBROUTINE READD (NC,NC,D)
C      INPUTS THE "D" MATRIX MEASUREMENT FEED-FORWARD DIST. MATRIX&.
C=====
      REAL*8 D(NC,NC),DUM,ANSR
      INTEGER IANS,I,J,K,L
      DATA I,Y,Z,N//
      WRITE (5,11C)
      DO 20 I=1,NC
      DO 20 J=1,NC
      WRITE (5,11C) I,J

```

```

100 CALL RDCREAL (ANSR)
110 D(I,J)=ANSF
120 CCNT INUE
130 CONT INUE
140
150 CALL FRTCMS ('CLRSCRN ')
160 WRITE (5,12C)
170 CALL MATPR1 (D,NO,NC)
180 WRITE (5,12C)
190 CALL RDCHAR (IANS)
200 IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
210 GC TO 60
220 WRITE (5,14C)
230 GO TO 40
240 CCNT INUE
250 IF (IANS.EQ.IZ) GC TC 90
260 WRITE (5,12C)
270 CALL RDCINT (IANS)
280 K=IANS
290 WRITE (5,16C)
300 CALL RDCINT (IANS)
310 L=IANS
320 WRITE (5,10C) K,L
330 CALL RDCREAL (ANSR)
340 CUM=ANSR
350 DO 80 I=1,NC
360 DO 70 J=1,NC
370 IF ((I.EQ.K).AND.(J.EQ.L)) D(I,J)=DUM
380 CCNT INUE
390 GO TO 30
400 CCNT INUE
410 CALL FRTCMS ('CLRSCRN ')
420 RETURN
430
440 FFORMAT (5X,14HTHE ELEMENT D(I2,1H,I2,2H)=)
450 FFORMAT (//,1X,54HTHE MEASUREMENT FEEDT+RUGH MATRIX / FEEDFOR
460 1WARD,/,5X,34H DISTRIBUTION MATRIX "D"-MATRIX&.,//,8X,4SHDIMENSION
470 2 = #,OBSERVATIONS NC& X # CONTROLS NC&)
480 FFORMAT (//,5X,50HTHE FEEDFORWARD DISTRIBUTION MATRIX "C"-MATRIX&
490 1.,//)
500 FFORMAT (//5X,54HDO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELE
510 1MENT?//,10X,19HTYPE "YES" OR "NO".)
520 FFORMAT (//,1X,51HTHWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
530 FFORMAT (5X,53HTHE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
540 1.,)
550 FFORMAT (5X,53HTHE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
560 1.,)
570 END

```

```

C=====OPT333610
SUBROUTINE READG (NS,NC,ISAG,G)OPT333620
C INTERACTIVELY INPUTS THE "G" MATRIX CONTROL DISTRIBUTION MATRIX&=OPT333630
C=====OPT333640
REAL*8 G(NS,NC),DUM,ANSR
INTEGER IANS,IJ,K,L,ISAG
DATA IY/,Y/,IZ/,N/,/
IF (ISAG.EQ.1) GO TO 40
WRITE (5,12C)
DO 20 I=1,NC
DO 10 J=1,NC
WRITE (5,11C) I,J
CALL RCREAL (ANSR)
G(I,J)=ANSR
CCNT INUE
CCNT INUE
1C-----
2C-----
3C-----
4C-----
5C-----
6C-----
7C-----
80-----
9C-----
100-----
C-----
CALL FRTCMS ('CLRSCRN ')
CCNT INUE
WRITE (5,13C)
CALL MATPRT (G,NS,NC)
WRITE (5,14C)
CALL RCCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60
GO TO 70
WRITE (5,15C)
GO TO 50
CCNT INUE
IF (IANS.EQ.IZ) GO TO 100
WRITE (5,16C)
CALL RCINT (IANS)
K=IANS
WRITE (5,17C)
CALL RDRINT (IANS)
L=IANS
WRITE (5,11C) K,L
CALL RCREAL (ANSR)
DUM=ANSR
DO 90 I=1,NC
DO 80 J=1,NC
IF ((I.EQ.K).AND.(J.EQ.L)) G(I,J)=DUM
CCNT INUE
GO TO 30
CCNT INUE
CALL FRTCMS ('CLRSCRN ')
RETURN
C-----

```



```

110 FORMAT (5X,14THE ELEMENT G(I2,1H,I2,2H)=) OPT34090
120 FORMAT (/5X,51HENTER THE CONTROL DISTRIBUTION MATRIX NC&) OPT34100
130 1.,//,10X,42HDIMENSION = # STATES NS&X# CONTRCLS "G"-MATRIX&... OPT34110
140 1//) OPT34120
150 FORMAT (/5X,54HDO YOU WISH TO CHANGE THE VALUE CF ANY MATRIX ELEMENT OPT34130
160 1ENT?//,10X,19HTYPE "YES" OR "NO".) OPT34140
170 FORMAT (1X,11HWARNING: IMPROPER DATA ENTRY ENTER "YES" CR "NO".) OPT34150
180 FORMAT (5X,5CHENTER THE ROW NUMBER OF THE ELEMENT TC BE CHANGED.) OPT34160
190 FORMAT (5X,53HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED OPT34170
200 1.) OPT34180
210 END OPT34190
220 OPT34200
230 SUBROUTINE READFB (NC,NS,FBGC) OPT34210
240 INPUTS THE "C" FEEDBACK GAIN CONTROL MATRIX&. OPT34220
250 REAL*8 FBGC(NC,NS),CUM,ANSR OPT34230
260 INTEGER IANS,I,J,K,L OPT34240
270 LATA IY,Y//,IZ//N// OPT34250
280 WRITE (5,11C) OPT34260
290 DO 20 I=1,NC OPT34270
300 DO 10 J=1,NS OPT34280
310 WRITE (5,11CC) I,J OPT34290
320 CALL RDREAL (ANSR) OPT34300
330 FBGC(I,J)=ANSR OPT34310
340 CCNT INUE OPT34320
350 CCNT INUE OPT34330
360 OPT34340
370 OPT34350
380 OPT34360
390 OPT34370
400 OPT34380
410 OPT34390
420 OPT34400
430 OPT34410
440 OPT34420
450 OPT34430
460 OPT34440
470 OPT34450
480 OPT34460
490 OPT34470
500 OPT34480
510 OPT34490
520 OPT34500
530 OPT34510
540 OPT34520
550 OPT34530
560 OPT34540
570 OPT34550
580 OPT34560

```



```

7C      DO 80 I=1,NC
8C      DO 70 J=1,NC
9C      IF ((I.EQ.K).AND.(J.EQ.L)) FBGC(I,J)=DUM
100     CONTINUE
110     GO TO 30
120     CONTINUE
130     CALL FRTCMS ('CLRSCRN ')
140     RETURN
150
160     -----
170     FORMAT (5X,14H THE ELEMENT C(I2,1H,I2,2H)=)
180     FORMAT (/ ,5X,52H ENTER THE FEEDBACK GAIN CCNTRL MATRIX "C"-MATRIX
190     180, //, 10X, 24H DIMENSION = # CONTROLS NCE X # STATES NSE.)
200     FORMAT (/ ,10X,45H THE FEEDBACK GAIN CONTROL MATRIX "C"-MATRIX & , / OPT34700
210     1)
220     FORMAT (//5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT
230     120? //, 10X,19H TYPE "YES" OR "NO".)
240     FORMAT (/ ,1X,11H WARNING: IMPROPER DATA ENTRY ENTER "YES" CR "NC".)
250     FORMAT (5X,52H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
260     FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
270     1.)
280     END
290
300     SUBROUTINE READAY (NC,ISAA,AY)
310     INPUTS THE "A" MATRIX DIAGONAL OUTPUT COST MATRIX &.
320
330     REAL*8 AY(NO,NO),DUM,ANSR
340     INTEGER IANS,I,J,K,L
350     DATA IY//Y//,IZ//N//
360     IF (ISAA.EQ.I) GO TO 30
370     WRITE (5,11C)
380     DO 20 I=1,NC
390     DO 10 J=1,NC
400     WRITE (5,12C) I,J
410     CALL RDRFAL (ANSR)
420     AY(I,J)=ANSR
430     CONTINUE
440
450     CALL FRTCMS ('CLRSCRN ')
460     WRITE (5,12C)
470     CALL MATPRT (AY,NO,NC)
480     WRITE (5,12C)
490     CALL RDCHAP (IANS)
500     IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
510     GO TO 60
520     WRITE (5,14C)
530     GO TO 40

```

```

6C      CONTINUE
        IF (IANS.EQ.1Z) GO TC 90
        WRITE (5,150)
        CALL RDINT (IANS)
        K=IANS
        WRITE (5,160)
        CALL RDINT (IANS)
        L=IANS
        WRITE (5,160) K,L
        CALL RDRREAL (ANSR)
        DUM=ANSR
        DO 80 I=1,NC
        DO 70 J=1,NC
        IF ((I.EQ.K).AND.(J.EQ.L)) AY(I,J)=DUM
        CONTINUE
        GO TO 30
7C      CGNT INUE
80      CALL FRTCMS ('CLRSCRN ')
9C      RETURN
C-----
100     FORMAT (5X,14H THE ELEMENT A(,I2,1H,I2,2H)=)
110     FORMAT (//,5X,54H ENTER THE OUTPUT MEASUREMENT CCST MATRIX "A"-MATO
1RX6.,/,5X,53H DIMENSION = # OBSERVATIONS NO& X # CESERVATIONS$ NO
28)
120     FORMAT (//,5X,50H THE OUTPUT MEASUREMENT COST MATRIX "A"-MATRIX&
1.,//)
130     FORMAT (//,5X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEME
1ENT?,/,10X,19H TYPE "YES" OR "NO".)
140     FORMAT (1X,51H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".)
150     FORMAT (5X,50H ENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED&
160     FORMAT (5X,53H ENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1.)
C=====
C===== SUBROUTINE READB (NC,ISAB,B)
C===== INPUTS THE "B" MATRIX CONTROL COST WEIGHTING MATRIX&
C=====
REAL*8 B(NC,NC),DUM,ANSR
INTEGER IANS,I,J,K,L
DATA I,Y/,I2/,N;/
IF (ISAB.EQ.1) GO TO 20
WRITE (5,90)
DO 10 I=1,NC
DO 10 J=1,NC
WRITE (5,80) I,J
CALL RDRREAL (ANSR)
B(I,J)=ANSR
1C

```


10	IF (IGAM.EQ.1) GO TO 40	OPT36010
20	WRITE (5,120)	OPT36020
30	DO 20 I=1,NS	OPT36030
40	DO 10 J=1,NG	OPT36040
50	WRITE (5,110) I,J	OPT36050
60	CALL RCDREAL (ANSR)	OPT36060
70	GAM(I,J)=ANSR	OPT36070
80	CONTINUE	OPT36080
90	CONTINUE	OPT36090
100	CALL FRTCMS ('CLRSCRN ')	OPT36100
110	CONTINUE	OPT36110
120	WRITE (5,130)	OPT36120
130	CALL MATPRT (GAM,NS,NG)	OPT36130
140	WRITE (5,140)	OPT36140
150	CALL RCDCHAR (IANS)	OPT36150
160	IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 60	OPT36160
170	GO TO 70	OPT36170
180	WRITE (5,150)	OPT36180
190	GO TO 50	OPT36190
200	CONTINUE	OPT36200
210	IF (IE.EQ.IZ) GO TO 100	OPT36210
220	WRITE (5,160)	OPT36220
230	CALL RCDINT (IANS)	OPT36230
240	K=IANS	OPT36240
250	WRITE (5,170)	OPT36250
260	CALL RCDINT (IANS)	OPT36260
270	L=IANS	OPT36270
280	WRITE (5,110) K,L	OPT36280
290	CALL RCDREAL (ANSR)	OPT36290
300	DUM=ANSR	OPT36300
310	DO 90 I=1,NS	OPT36310
320	DO 80 J=1,NG	OPT36320
330	IF ((I.EQ.K).AND.(J.EQ.L)) GAM(I,J)=DUM	OPT36330
340	CONTINUE	OPT36340
350	GO TO 30	OPT36350
360	CONTINUE	OPT36360
370	GO TO 30	OPT36370
380	CONTINUE	OPT36380
390	CALL FRTCMS ('CLRSCRN ')	OPT36390
400	RETURN	OPT36400
410	FORMAT (5X,16H THE ELEMENT GAM(I2,IH,I2,2H)=)	OPT36410
420	FOR "GAMMA",26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36420
430	1X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36430
440	2X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36440
450	1X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36450
460	1X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36460
470	1X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36470
480	1X "GAMMA", 26H ENTER THE PROCESS NOISE DISTRIBUTION, /, 5X, 24H MATR	OPT36480


```

1ENT?//,10X,19HTYPE "YES" OR "NO".)
1EQ  FORMAT (1X,19HWARNING: IMPROPER DATA ENTRY
16C  FORMAT (5X,19HENTER THE ROW NUMBER OF THE ELEMENT TO BE CHANGED.)
17C  FORMAT (5X,19HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1.)
1END
C=====
C  SUBROUTINE READQ (NG,Q)
C  INTERACTIVELY INPUTS THE "Q" MATRIX NOISE WEIGHTING MATRIX&
C=====
REAL*8  G(NG,NG),DUM,ANSR
INTEGER IANS,I,J,K,L
DATA IY/,Y/,IZ/,N;/
WRITE (5,11C)
DO 20 I=1,NG
  DO 10 J=1,NG
    WRITE (5,12C) I,J
    CALL RCREAL (ANSR)
    Q(I,J)=ANSR
  CCNT INUE
CCNT INUE
1C  CALL FRTCMS ('CLRSCRN ')
2C  WRITE (5,12C)
3C  CALL MATPRT (Q,NG,NG)
4C  WRITE (5,13C)
5C  CALL RCCHAR (IANS)
6C  IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 50
  GO TO 60
  WRITE (5,14C)
  GO TO 40
  CCNT INUE
  IF (IANS.EQ.IZ) GO TC 90
  WRITE (5,15C)
  CALL RCINT (IANS)
  K=IANS
  WRITE (5,16C)
  CALL RCINT (IANS)
  L=IANS
  WRITE (5,17C) K,L
  CALL RDRREAL (ANSR)
  DUM=ANSR
  DO 90 I=1,NG
    DO 70 J=1,NG
      IF ((I.EQ.K).AND.(J.EQ.L)) Q(I,J)=DUM
    CCNT INUE
  CCNT TO 30
  GO TO 30

```



```

SC      CONTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----
100     FORMAT (5X,14H THE ELEMENT Q(I,12,1H,I2,2H)=)
110     FCRMAT (//,15X,44H ENTER THE PROCESS NOISE PSD WEIGHTING MATRIX,/,5X OPT37020
      1,12H "Q" MATRIX,/,15X,42H DIMENSION=# NOISE SOURCES NG& OPT37030
      2,X,/,17X,27H #PROCESS NOISE SOURCES NG& OPT37040
120     FORMAT (//,15X,42H THE PROCESS NOISE WEIGHTING MATRIX,/,15X,42H THE VALUE OF ANY MATRIX ELEMENT OPT37050
130     FORMAT (//,15X,54H DO YOU WISH TO CHANGE THE VALUE OF ANY MATRIX ELEMENT OPT37060
      1ENT?,/,10X,19H TYPE "YES" OR "NO".) OPT37070
140     FORMAT (//,15X,44H ENTER "YES" OR "NO".) OPT37080
150     FCRMAT (5X,14H WARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".) OPT37090
160     FORMAT (5X,14H WARNING: THE ROW NUMBER OF THE ELEMENT TO BE CHANGED OPT37100
      1.) OPT37110
      END OPT37120
C===== OPT37130
      SUBROUTINE READR (NG,RC) OPT37140
C      INTERACTIVELY INPUTS THE "R" MATRIX= OPT37150
C      MEASUREMENT NOISE DISTRIBUTION MATRIX. OPT37160
C===== OPT37170
      REAL*8 RC(NO,NO),DUM,ANSR OPT37180
      INTEGER IANS,I,J,K,L OPT37190
      DATA IY,Y,Y,I,Z,N;/ OPT37200
      WRITE (5,90) OPT37210
      DO 10 I=1,NC OPT37220
      DO 10 J=1,NC OPT37230
      WRITE (5,80) I,J OPT37240
      CALL RDCREAL (ANSR) OPT37250
      RC(I,J)=ANSR OPT37260
1C----- OPT37270
2C      CALL FRTCMS ('CLRSCRN ') OPT37280
      WRITE (5,100) OPT37290
      CALL MATPRT (RC,NO,NC) OPT37300
      WRITE (5,110) OPT37310
      CALL RDCCHAR (IANS) OPT37320
      IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 40 OPT37330
      GO TO 50 OPT37340
4C      WRITE (5,120) OPT37350
      GC TO 30 OPT37360
5C      CONTINUE OPT37370
      IF (IANS.EC.IZ) GO TC 70 OPT37380
      WRITE (5,130) OPT37390
      CALL RDCINT (IANS) OPT37400
      K=IANS OPT37410
      WRITE (5,140) OPT37420
      CALL RDCINT (IANS) OPT37430
      L=IANS OPT37440

```



```

60      GO TO 40
      CCNTINUE
      IF (IANS.EQ.1Z) GO TC 90
      WRITE (5,16C)
      CALL RCINT (IANS)
      K=IANS
      WRITE (5,16C)
      CALL RDINT (IANS)
      L=IANS
      WRITE (5,10C) K,L
      CALL RDRAL (ANSR)
      DUM=ANSR
      DO 80 I=1,NS
      DO 70 J=1,NC
      IF ((I.EQ.K).AND.(J.EQ.L)) FBGE(I,J)=DUM
      CCNTINUE
      GO TO 30
      CCNTINUE
      CALL FRTCMS ('CLRSCRN ')
      RETURN
C-----
100  FORMAT (5X,14HTHE ELEMENT K(I2,1H,I2,2H)=)
110  FORMAT (//5X,54HENTER THE FEEDBACK GAIN ESTIMATOR MATRIX "K"-MATR
120  1IX6,/,10X,48HDIMENSION=# STATES NS&X# CBSERVATICS NC&.)
120  1//)
120  FORMAT (//5X,54HDC YOU WISH TO CHANGE THE VALUE CF ANY MATRIX ELE
140  1MENT?//10X,19HTYPE "YES" OR "NO".)
140  FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" CR "NO".)
150  FORMAT (5X,52HENTER THE ROW NUMBER OF THE ELEMENT TC BE CHANGED.)
160  FORMAT (5X,52HENTER THE COLUMN NUMBER OF THE ELEMENT TO BE CHANGED
1)
C=====
C      SUBROUTINE READW (NG,WR)
C      INTERACTIVELY INPUTS THE "W0" MATRIX STEADY DISTURBANCE VECTOR =
C      MATRIX& ELEMENT BY ELEMENT.
C=====
      REAL*8 WR(NG),DUM,ANSR
      INTEGER IANS,I,K
      DATA IY,Y,IZ/'N' /
      WRITE IY(5,1CC)
      DO 10 I=1,NC
      WRITE I(5,80) I
      CALL RDRAL (ANSR)
      WR(I)=ANSR
      CCNTINUE
1C

```



```

CCOUNT=COUNT+1
IF (COUNT.LT.3) GO TC 20
WRITE (5,6C)
GO TO 40
CCNTINUE
READ (5,*,END=30,ERR=30) ANSR
RETURN 5
REWIND 5
WRITE (5,50)
GO TO 10
CCNTINUE
STOP
C-----
50  FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.)
60  FORMAT (//,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED )
END
C=====
C  SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY
C  INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS AN IMPROPER
C  DATA CHARACTER THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
SUBROUTINE RDINT (IANS)
INTEGER CCOUNT,IANS
C-----
10  CCOUNT=0
CCNTINUE
CCOUNT=COUNT+1
IF (COUNT.LT.3) GO TC 20
WRITE (5,6C)
GO TO 50
CCNTINUE
READ (5,*,END=40,ERR=40) IANS
IF (IANS) 40,40,30
CCNTINUE
RETURN 5
REWIND 5
WRITE (5,70)
GO TO 10
CCNTINUE
STOP
C-----
60  FORMAT (//,5X,49HPROGRAM TERMINATION - TWO IMPROPER DATA ENTRIES
1)
70  FORMAT (1X,56HWARNING:  IMPROPER DATA ENTRY  ENTER A POSITIVE INTEGER.)
END
C=====

```



```

C SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY      =OPT393370
C (YES, OR NO) INTO A FCRTRAN PROGRAM. IF THE USER INADVERTENTLY      =OPT393380
C ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=OPT393390
C=====
C SUBROUTINE FDCCHAR (IANS)      =OPT393400
C INTEGER COUNT,IANS            =OPT393410
C DATA IY/Y,Y/,IZ/N./         =OPT393420
C-----
C COUNT=0                       =OPT393430
C COUNT=COUNT+1               =OPT393440
C IF (COUNT.LT.3) GO TC 20     =OPT393450
C WRITE (5,6C)                 =OPT393460
C GO TO 40                     =OPT393470
C CONTINUE                     =OPT393480
C READ (5,70,END=30,ERR=30) IANS =OPT393490
C RETURN 5                     =OPT393500
C REWIND 5                     =OPT393510
C GO TO 10                     =OPT393520
C CONTINUE                     =OPT393530
C WRITE (5,50)                 =OPT393540
C GO TO 10                     =OPT393550
C CONTINUE                     =OPT393560
C STOP                         =OPT393570
C-----
C FORMAT (1X,60HWARNING: NULL STRINGS ARE NOT ALLOWED, ENTER "YES"      =OPT393580
C 10P "NO".)                   =OPT393590
C FORMAT (///,5X,47HPROGRAM TERMINATION - TWO NULL STRINGS ENTERED ) =OPT393600
C FORMAT (A1)                  =OPT393610
C END                           =OPT393620
C=====
C SUBROUTINE MATPRT -- DISPLAYS A TWO-DIMENSIONAL ARRAY (16 COLS. MAX)=OPT393630
C IN VARIABLE SCREEN FORMAT FOR USER EASE IN ROW IDENTIFICATION.      =OPT393640
C=====
C SUBROUTINE MATPRT (PRTT,NROW,NCOL) =OPT393650
C IMPLICIT REAL*8 (A-H,O-Z)      =OPT393660
C DIMENSION PRTT(NROW,NCOL)     =OPT393670
C-----
C IF (NCOL.EC.0) NCOL=1          =OPT393680
C IF (NCOL.EC.1) WRITE (5,10)   =OPT393690
C IF (NCOL.EC.2) WRITE (5,20)   =OPT393700
C IF (NCOL.EC.3) WRITE (5,30)   =OPT393710
C IF (NCOL.EC.4) WRITE (5,40)   =OPT393720
C IF (NCOL.EC.5) WRITE (5,50)   =OPT393730
C IF (NCOL.EC.6) WRITE (5,60)   =OPT393740
C IF (NCOL.EC.7) WRITE (5,70)   =OPT393750
C IF (NCOL.EC.8) WRITE (5,80)   =OPT393760
C IF (NCOL.EC.9) WRITE (5,90)   =OPT393770
C IF (NCOL.EC.10) WRITE (5,100) =OPT393780
C IF (NCOL.EC.11) WRITE (5,110) =OPT393790
C IF (NCOL.EC.12) WRITE (5,120) =OPT393800
C IF (NCOL.EC.13) WRITE (5,130) =OPT393810
C IF (NCOL.EC.14) WRITE (5,140) =OPT393820
C IF (NCOL.EC.15) WRITE (5,150) =OPT393830
C IF (NCOL.EC.16) WRITE (5,160) =OPT393840

```



```

1,/,5X, 'RUN TO BE USED IN THIS RUN?'.//,5X,39HNOTE: THE MOP T4 1290
2ATRIX,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1300
3SERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1310
4INDIVIDUAL MATPIX ELEMENTS,/,15X,19HTYPE "YES" OR "NO".} OPT T4 1320
FORMAT (//,/,5X,52HDO YOU WISH TO SAVE THE "GAMMA"--MATRIX FROM THE OPT T4 1330
1LAST,/,5X, 'WILL BE REDISPLAYED AT,/,5X,34HTHE PROPER INPUT SEQUENCE? T9PT4 1340
2THE INTERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1350
3INDIVIDUAL MATPIX ELEMENTS,/,15X,19HTYPE "YES" OR "NO".} OPT T4 1360
4FORMAT (//,/,5X,48HDO YOU WISH TO SAVE THE "A"--MATRIX FROM THE LAST OPT T4 1370
1,/,5X, 'RUN TO BE USED IN THIS RUN?'.//,5X,39HNOTE: THE MOP T4 1380
2ATRIX,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1390
3SERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1400
4INDIVIDUAL MATPIX ELEMENTS,/,15X,19HTYPE "YES" OR "NO".} OPT T4 1410
FORMAT (//,/,5X,48HDC YCU WISH TO SAVE THE "B"--MATRIX FROM THE LAST OPT T4 1420
1,/,5X, 'RUN TO BE USED IN THIS RUN?'.//,5X,39HNOTE: THE MOP T4 1430
2ATRIX,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1440
3SERVAL,/,5X,40HAND YOU WILL HAVE THE OPTION OF CHANGING,/,5X,27HINDO T4 1450
4INDIVIDUAL MATPIX ELEMENTS,/,15X,19HTYPE "YES" OR "NO".} OPT T4 1460
FORMAT (1X,51HWARNING: IMPROPER DATA ENTRY ENTER "YES" OR "NO".) OPT T4 1470
END OPT T4 1480
OPT T4 1490
C===== OPT T4 1500
C SUBROUTINE RDMAT -- READS THE F, G, H, GAM, A AND B MATRICES FROM =OPT T4 1510
C MATRICES FROM THE DATA FILE OPTMAT ON FILEDEF 9. =OPT T4 1520
C===== OPT T4 1530
SUBROUTINE RDMAT(BA,G,H,GAM,FBGC,FBGE,AY,B,NS,NC,NO,NG,IRDMAT) OPT T4 1540
IMPLICIT REAL*8(A-H,C-Z) OPT T4 1550
DIMENSION BA(NS,NS),G(NS,NC),HO(NO,NS),GAM(NS,NG),FBGC(NC,NS), OPT T4 1560
1AY(NO,NO),B(NC,NC),FBGE(NS,NO) OPT T4 1570
IF(IRDMAT.EC.0) RETURN OPT T4 1580
REWIND 9 OPT T4 1590
READ (9,20) K,I,ANS OPT T4 1600
READ (9,20) NSI,NCI,NOI,NGI OPT T4 1610
READ (9,10) ((BA(I,J),J=1,NSI),I=1,NSI) OPT T4 1620
READ (9,10) ((G(I,J),J=1,NCI),I=1,NSI) OPT T4 1630
READ (9,10) ((HO(I,J),J=1,NGI),I=1,NOI) OPT T4 1640
READ (9,10) ((GAM(I,J),J=1,NSI),I=1,NSI) OPT T4 1650
READ (9,10) ((FBGC(I,J),J=1,NCI),I=1,NCI) OPT T4 1660
READ (9,10) ((FBGE(I,J),J=1,NOI),I=1,NSI) OPT T4 1670
READ (9,10) ((AY(I,J),J=1,NOI),I=1,NCI) OPT T4 1680
READ (9,10) ((B(I,J),J=1,NCI),I=1,NCI) OPT T4 1690
RETURN OPT T4 1700
----- OPT T4 1710
10 FCORMAT(4(D2C.13)) OPT T4 1720
20 FCORMAT(4I5) OPT T4 1730
END OPT T4 1740
C===== WRITES THE F, G, HO & GAM MATRICES TO ===== OPT T4 1750
C SUBROUTINE WRDMAT -- WRITES THE F, G, HO & GAM MATRICES TO ===== OPT T4 1760

```


APPENDIX C
OPTGRAPH PROGRAM LISTING

This portion of the thesis contains the OPTGRAPH FORTRAN program (54 pages).

```

OPT000010
OPT000020
OPT000030
OPT000040
OPT000050
OPT000060
OPT000070
OPT000080
OPT000090
OPT000100
OPT000110
OPT000120
OPT000130
OPT000140
OPT000150
OPT000160
OPT000170
OPT000180
OPT000190
OPT000200
OPT000210
OPT000220
OPT000230
OPT000240
OPT000250
OPT000260
OPT000270
OPT000280
OPT000290
OPT000300
OPT000310
OPT000320
OPT000330
OPT000340
OPT000350
OPT000360
OPT000370
OPT000380
OPT000390
OPT000400
OPT000410
OPT000420
OPT000430
OPT000440
OPT000450
OPT000460
OPT000470
OPT000480

      OPTGRAPH
      BY MICHAEL H. LAPTAS

      THIS PROGRAM IS AN INTERACTIVE DISSPLA GRAPHICS
      ROUTINE USED TO PLOT POLE-ZERO, ROOT-LOCUS, BODE,
      NYQUIST, AND NICHOLS PLOTS FROM THE OPTSYS PROGRAM
      OPEN LOOP TRANSFER FUNCTIONS.

      15 AUG 1984

      IMPLICIT REAL*8(A-H,C-Z)
      REAL*8 MA,MAG
      REAL*4 XDO,YDO,XNO,YNO,GOL
      COMPLEX*16 FTN,PLN,RTC,PLD,WN
      INTEGER I,J,LL,KK,NC,NS,NOB,IPZ,IRTLO,IBOD,II,JJ,LINES,LINE,K1,K2
      *      ,NN,IE,IA,IB
      COMMON /GRAFF/ OLN(99,2,12,12),OLD(99,2),GAN(2,12,12),CO
      COMMON /SGRAF/ XDO(500),YDO(500),XNO(500),YNO(500)
      COMMON /SHEAD/ LINES(4,18),LINE(9)
      COMMON /FFREQ/ RTN(99),PLN(99),PLD(99),WN(500),QI(500),
      *      CR(500),QPSR(500),MAG(500)
      COMMON /SCALE/ TS,TH,TH1,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10

      SUPPRESS POSSIBLE UNDERFLOW AND OVERFLOW ERROR MESSAGES-SUMMARY ONLY=
      CALL ERRSET (207,256,-1,1,1,209)
      *      ZERO VARIABLES
      NS = 0
      NOB = 0
      DO 5 J = 1,2
      DO 5 J = 1,99
      CLD(J,I) = C.0
      DO 10 I = 1,12
      DO 10 J = 1,12
      DO 10 KK = 1,299
      DO 10 LL = 1,99
      CLN(LL,KK,J,I) = 0.0
      DO 20 I = 1,12
      DO 20 J = 1,12
      DO 20 KK = 1,12
      GANI(KK,J,I) = 0.0
      *      SELECT TRANSFER FUNCTION TYPE

```

```

3C      CALL FRTCMS('CLRSCRN ')
        WRITE(6,92C)
        WRITE(6,750)
35      WRITE(6,76C)
        CALL RCINT(IANS)
        IF(IANS.GT.4000.IANS.LT.1) GO TO 34
        GO TO 36
34      WRITE(6,505)
        GO TO 35
36      CCNTINUE
        IF(IANS.EQ.1) IFL = 10
        IF(IANS.EQ.2) IFL = 1
        IF(IANS.EQ.3) IFL = 4
        IF(IANS.EQ.4) GO TO 95
C-----READ(IFL,10C) IA,IB,NS,NC,NOB,IE,ITFX INPUT DISK-----
C-----CHECK DATA FILE FOR PROPER DATA-----
        IF(IB.NE.0) GO TO 41
        IF(ITFX.EQ.1) WRITE(6,510)
        IF(ITFX.EQ.2) WRITE(6,511)
        IF(ITFX.EQ.3) WRITE(6,512)
        GO TO 35
41      IF(NS.NE.0) GO TO 42
        WRITE(6,513)
        GO TO 35
42      IF(NC.NE.0) GO TO 43
        IF(ITFX.EQ.1) WRITE(6,515)
        IF(ITFX.EQ.2) WRITE(6,517)
        IF(ITFX.EQ.3) WRITE(6,516)
        GO TO 35
43      IF(NOB.NE.0) GO TO 44
        IF(ITFX.EQ.1) WRITE(6,516)
        IF(ITFX.EQ.2) WRITE(6,516)
        IF(ITFX.EQ.3) WRITE(6,515)
        GO TO 35
C-----MARKCV CRITERIA FOR DETERMINING EXTRANEOUS ZEPES-----
44      IF(IE.EQ.0) IE = 6
        CC = 10.**(-IE)
C-----DENOMINATOR-----
        CC 50 I = 1,NS
        CC READ(IFL,120) OLD(I,1),OLD(I,2)
50      CCNTINUE
C-----NUMERATOR / GAIN / ORDER-----
        NN = NOB*NC
        CC 90 I = 1,NN
        READ(IFL,101) KK,K2,K1
        READ(IFL,12C) GAN(I,K2,K1),GAN(2,K2,K1)
        IF(GAN(1,K2,K1).NE.0.0) GO TO 72

```



```

CALL YNAME(' $',100)
CALL GRAF(XMIN,SCALE,XMAX,YMIN,SCALE,YMAX)
XL = (XMAX-XMIN)*0.1
YL = (YMAX-YMIN)*0.1
CALL RLVEC(XMAX+XL,0.0,XMIN-XL,0.0,0000)
CALL RLVEC(C.0,YMAX+YL,0.0,YMIN-YL,0000)
C-----HEADING-----
NN = NL + 1
IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TRANSFER POLE-ZERO MAP$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TRANSFER POLE-ZERO MAP$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TRANSFER POLE-ZERC MAP$',100,T3,NN)
C-----POLE-----
CALL MARKER(4)
CALL SCLPIC(2.)
CALL CURVE(XDO,YDO,NS,-1)
C-----ZERO-----
IF(J1.EQ.0) GO TO 36
CALL MARKER(16)
CALL SCLPIC(2.)
CALL CURVE(XNO,YNO,J1,-1)
36 CONTINUE
C-----LEGEND-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAGE('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAGE('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAGE('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----GRID-----
CALL DCT

```



```

CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(0)
IF(IPTR.NE.2) GO TO 50
CALL FRTCMS('CLRSCRN ')
WRITE(6,130)
-----CHANGES TO PLOT-----
5C
CCNT INUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
WRITE(6,100)
WRITE(6,110)
CALL RDINT(IANS)
IF(IANS.GT.500,IANS.LT.1) GO TO 15
GO TO 25
15 WRITE(6,510)
GO TO 30
25 CCNT INUE
ICHG = 1
GO TO (40,50,60,70,200),IANS
-----TABULAR DATA-----
202 CCNT INUE
J1 = 0
CRD = 0
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
CALL PRTR(IPT)
CALL FRTCMS('CLRSCRN ')
WRITE(IPT,120)
IF(ITFX.EQ.1) WRITE(IPT,710)
IF(ITFX.EQ.2) WRITE(IPT,711)
IF(ITFX.EQ.3) WRITE(IPT,712)
WRITE(IPT,300) K1
WRITE(IPT,301) K2
WRITE(IPT,320) GAN(2,K2,K1)
WRITE(IPT,311) NS
WRITE(IPT,340)
DO 210 I = 1, NS
21C WRITE(IPT,330) OLD(I,1),OLD(I,2)
IF(ORD.EQ.0) GO TO 230

```

```

WRITE(IPT,310) ORD
WRITE(IPT,350)
WRITE(IPT,340)
DO 220 I = 1, NS
IF ((DABS(OLN(I,1,K2,K1)).LE.CO).AND.(DABS(OLN(I,2,K2,K1)).LE.CO)
*) GO TO 22C
J1=J1+1
WRITE(IPT,330) CLN(I,1,K2,K1),OLN(I,2,K2,K1)
CONTINUE - J1
IF (IZR.LE.C) GO TO 240
DO 225 I = 1, IZR
WRITE(IPT,330) ZERO, ZERO
GO TO 240
WRITE(IPT,350)
WRITE(IPT,380)
C-----CHANGES FOR TABULAP DATA-----
240 IF (IPT.EQ.2) WRITE (6,401)
IF (IPT.EQ.2) WRITE (6,402)
WRITE(6,400)
CALL RSCHAR (IANS)
IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
GO TO 250
WRITE (6,500)
GO TO 240
245 CCNTINUE
IF (IANS.EC.IY) GO TO 202
IF (IANS.EC.IZ) GO TO 200
C-----
8C RETURN
C-----
100 FORMAT (//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES TO:',//,10X,'1.',
*LOTTER / PAGE SIZE',//,10X,'2. GRAPH LIMITS',//,10X,'3. HEADING',//,10X,'4. NC ADDITIONAL CHANGES -- PLOT POLE-ZERO MAP',//,10X,'5.
*CHANGES -- EXIT POLE-ZERO PLOTTING ROUTINE',//)
110 FORMAT (//,5X,'ENTER OPTION NUMBER',//)
120 FORMAT (//,15X,'POLE-ZERO MAP')
130 FORMAT (//,15X,'A DISSPLA METAFILE HAS BEEN CREATED',//,
*15X,'CLEAR SCREEN TO CONTINUE',//,15X,'15)
FORMAT (//,10X,'INPUT NUMBER =',I5)
FORMAT (//,10X,'NUMERATOR ORDER =',I5)
FORMAT (//,10X,'DENOMINATOR ORDER =',I5)
FORMAT (//,10X,'TRANSFER FUNCTION (DC) GAIN =',D12.4,/)
FORMAT (//,10X,'REAL LOCATIONS')
FORMAT (//,10X,'IMAGINARY PART',/)
FORMAT (//,10X,'ZERO LOCATIONS')
FORMAT (//,10X,'POLE LOCATIONS')

```



```

41 IF(ITFX.EQ.2) WRITE(6,711)
   IF(ITFX.EQ.3) WRITE(6,712)
   CALL GSETUP(IPTR,PAGEX,PAGEY)
50 IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(.CLRSCRN)
55 CALL GGRAF(XMIN,XMAX,YMIN,YMAX)
   IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(.CLRSCRN)
60 IF(ICHG.EQ.1) GO TO 90
   CALL FRTCMS(.CLRSCRN)
70 IF(ICHG.EQ.1) GO TO 90
   CALL GHEAD(NL)
   CONTINUE
   IF(IPTR.EQ.2) GO TO 10
C-----TEK 618-----
   CALL TEK618
   CALL PAGE(PAGEX,PAGEY)
   CALL HPROT('AUTO')
   CALL HWSCAL('SCREEN')
   CALL NCBDRF
   GO TO 20
C-----COMPRS-----
10 CONTINUE
   CALL COMPRS
   CALL PAGE(PAGEX,PAGEY)
20 CONTINUE
C-----SCALING CONSTANTS-----
   CALL PSCALE(PAGEX)
C-----PLOT EXECUTION-----
   CALL NOCHECK
   CALL GRACE(C)
   CALL PHYSOR(T1,T1)
   XAXIS = PAGEX - 2.0*T1
   YAXIS = PAGEY - T1 - T2
   CALL AREA2C(XAXIS, YAXIS)
   CALL TRIPLX
   CALL HEIGHT(TH)
   CALL XNAME(' ',100)
   CALL HEIGHT(TH)
   CALL YNAME(' ',100)
   CALL GRAF(XMIN,SCALE,XMAX,YMIN,SCALE,YMAX)
   XL = (XMAX-XMIN)*0.1
   YL = (YMAX-YMIN)*0.1
   CALL RLVEC(C,XL,0.0,XMIN-XL,0.0,0.0)
   CALL FLVEC(C,YL,0.0,YMAX+YL,0.0,YMIN-YL,0.0)
C-----HEADING-----
   NN = NL + 1

```

```

OPT05290
OPT05300
OPT05310
OPT05320
OPT05330
OPT05340
OPT05350
OPT05360
OPT05370
OPT05380
OPT05390
OPT05400
OPT05410
OPT05420
OPT05430
OPT05440
OPT05450
OPT05460
OPT05470
OPT05480
OPT05490
OPT05500
OPT05510
OPT05520
OPT05530
OPT05540
OPT05550
OPT05560
OPT05570
OPT05580
OPT05590
OPT05600
OPT05610
OPT05620
OPT05630
OPT05640
OPT05650
OPT05660
OPT05670
OPT05680
OPT05690
OPT05700
OPT05710
OPT05720
OPT05730
OPT05740
OPT05750
OPT05760

```



```

IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LIME,100,T3,NN)
35 CCNTINUE
IF(ITFX.EQ.1)
* CALL HEADIN('ROOT-LOCUS PLOT (OPEN LOOP TF)$',100,T3,NN)
* IF(ITFX.EQ.2)
* CALL HEADIN('ROOT-LOCUS PLCT (NOISE TF)$',100,T3,NN)
* IF(ITFX.EQ.3)
* CALL HEADIN('ROOT-LCCUS PLOT (CCOMPENSATOR TF)$',100,T3,NN)
C-----INPUT ARRAYS-----
ZERO = 0.0
J1 = 0
ND1 = NS + 1
ND2 = NS + 2
DC 5 LL = 1,NS
XNO(LL) = C.O
YNO(LL) = C.O
QI(LL) = 0.C
QR(LL) = 0.C
XDO(LL) = SNGL(OLD(LL,1))
YDO(LL) = SNGL(OLD(LL,2))
RTD(LL) = CCPLX(OLD(LL,1),OLD(LL,2))
RTN(LL) = CCPLX(ZERC,ZERO)
IF((DABS(OLN(LL,1,K2,K1)).LE.CO).AND.(DABS(OLN(LL,2,K2,K1)).LE.CO)
*) GO TO 6
J1 = J1 + 1
XNO(J1) = SNGL(OLN(LL,1,K2,K1))
YNO(J1) = SNGL(OLN(LL,2,K2,K1))
RTN(J1) = CCPLX(OLN(LL,1,K2,K1),OLN(LL,2,K2,K1))
CCNTINUE
ORD = DINT(GAN(1,K2,K1))
IF(ORD.GT.J1) J1 = ORD
GOL = SNGL(GAN(2,K2,K1))
GAIN = GAN(2,K2,K1)
C-----POL E-----
CALL MARKER(4)
CALL SCLPIC(2)
CALL CURVE(XDO,YDO,NS,-1)
C-----ZERO-----
IF(J1.EQ.0) GO TO 36
CALL MARKER(16)
CALL SCLPIC(2)
CALL CURVE(XNO,YNO,J1,-1)
36 CCNTINUE
C-----POLYNOMIALS-----

```


C	-----	DENOMENATOR	-----	OPT06250
	CALL MAKPOL(NS,RTD,PLD)			OPT06260
	GI(1) = 1.0			OPT06270
7	DO 7 I = 2,ND1			OPT06280
C	QI(I) = DREAL(PLD(ND2-I))			OPT06290
	-----	NUMERATOR	-----	OPT06300
	IF(J1.EQ.0) GO TO 9			OPT06310
	CALL MAKPOL(J1,RTN,PLN)			OPT06320
	JJ = NS - J1 + 2			OPT06330
	QR(JJ-1) = 1.0			OPT06340
	DO 8 I = JJ,ND1			OPT06350
8	QR(I) = DREAL(PLN(ND2-I))			OPT06360
	GO TO 11			OPT06370
9	QR(ND1) = 1.0			OPT06380
	QR(1) = 0.0			OPT06390
C	-----	ROOT FINDING ROUTINE	-----	OPT06400
11	KIC = (KMAX-KMIN)/2000.			OPT06410
	K = KMIN			OPT06420
	DO 52 J = 1,2000			OPT06430
	DO 37 I = 1,ND1			OPT06440
	MAG(II) = GI(II) + K*GAIN*QR(II)			OPT06450
37	CONTINUE			OPT06460
	CALL ZRPOLY(MAG,NS,WN,IER)			OPT06470
	DO 51 II = 1,NS			OPT06480
	XR = DREAL(WN(II))			OPT06490
	XNO(II) = SNGL(XR)			OPT06500
	YI = DIMAG(WN(II))			OPT06510
	YNO(II) = SNGL(YI)			OPT06520
51	CCNT = INUE			OPT06530
	CALL MARKER(4)			OPT06540
	CALL SCLPIC(1.)			OPT06550
	CALL CURVE(XNO,YNO,NS,-1)			OPT06560
	K=K+KIC			OPT06570
52	CONTINUE			OPT06580
C	-----	LEGEND	-----	OPT06590
	XX = XAXIS - T7			OPT06600
	YY = YAXIS + T8			OPT06610
	CALL HEIGHT(TH1)			OPT06620
	CALL MESSAGE(,INPUT # = \$,100,XX+T8,YY+T10)			OPT06630
	CALL HEIGHT(TH1)			OPT06640
	CALL INTNO(K1,ABUT,ABUT)			OPT06650
	CALL HEIGHT(TH1)			OPT06660
	CALL MESSAGE(,OUTPUT # = \$,100,XX+T8,YY+T5)			OPT06670
	CALL HEIGHT(TH1)			OPT06680
	CALL INTNO(K2,ABUT,ABUT)			OPT06690
	CALL HEIGHT(TH1)			OPT06700
	CALL MESSAGE(,DC GAIN = \$,100,XX+T8,YY+T8)			OPT06710
	CALL HEIGHT(TH1)			OPT06720

C-----	CALL REALNC(GOL,-3,'ABUT','ABUT')	OPT06730
	CALL BLREC(XX,YY,T5,T6,0.02)	OPT06740
		OPT06750
	CALL DOT	OPT06760
	CALL GRID(1,1)	OPT06770
	CALL RESET('DOT')	OPT06780
	CALL ENDPL(C)	OPT06790
	IF(IPTR.NE.2) GO TO 90	OPT06800
	CALL FRTCMS('CLRSCRN')	OPT06810
	WRITE(6,130)	OPT06820
C-----		OPT06830
3C	CONTINUE	OPT06840
	CALL FRTCMS('CLRSCRN')	OPT06850
	WRITE(6,120)	OPT06860
	IF(ITFX.EQ.1) WRITE(6,710)	OPT06870
	IF(ITFX.EQ.2) WRITE(6,711)	OPT06880
	IF(ITFX.EQ.3) WRITE(6,712)	OPT06890
	WRITE(6,1100)	OPT06900
30	WRITE(6,1110)	OPT06910
	CALL PCINT(IANS)	OPT06920
	IF(IANS.GT.600R.IANS.LT.1) GO TO 15	OPT06930
	GO TO 25	OPT06940
15	WRITE(6,510)	OPT06950
	GO TO 30	OPT06960
25	CONTINUE	OPT06970
	ICHG = 1	OPT06980
	GO TO (40,50,60,55,70,200),IANS	OPT06990
C-----		OPT07000
202	CONTINUE	OPT07010
	ICHG = 0	OPT07020
251	CALL FRTCMS('CLRSCRN')	OPT07030
	IF(ICHG.EQ.1) GO TO 249	OPT07040
	WRITE(6,120)	OPT07050
	IF(ITFX.EQ.1) WRITE(6,710)	OPT07060
	IF(ITFX.EQ.2) WRITE(6,711)	OPT07070
	IF(ITFX.EQ.3) WRITE(6,712)	OPT07080
249	CALL IPTRTR(IFT)	OPT07090
	IF(ICHG.EQ.1) GO TO 290	OPT07100
252	CALL FRTCMS('CLRSCRN')	OPT07110
	CALL RANGK(KMIN,KMAX)	OPT07120
	IF(ICHG.EQ.1) GO TO 290	OPT07130
253	CALL FRTCMS('CLRSCRN')	OPT07140
	CALL NPPTS(PTS)	OPT07150
	IF(ICHG.EQ.1) GO TO 290	OPT07160
C-----		OPT07170
254	CONTINUE	OPT07180
	ZERO = 0.0	OPT07190
	J1 = 0	OPT07200


```

IF(J1.EQ.0) GO TO 209
CALL MAKPOL(J1,RTN,PLN)
JJ = NS - J1 + 2
QP(JJ-1) = 1.0
CD 208 I = JJ,ND1
QR(I) = CREAL(PLN(ND2-I))
GO TO 211
QR(ND1) = 1.0
QR(1) = 0.0
C-----ROOT FINDING ROUTINE-----
211 WRITE(IPT,220)
KIC = (KMAX-KMIN)/DFLCAT(PTS)
K = KMIN
DO 262 J = 1,PTS
DO 237 II = 1,ND1
MAG(II) = CI(II) + K*GAIN*QR(II)
CCNT INUE
CALL ZRPOLY(MAG,NS,KN,IER)
WRITE(IPT,221) K
WRITE(IPT,240)
DO 261 II = 1,NS
XR = DREAL(WN(II))
YI = DIMAG(WN(II))
WRITE(IPT,220) XR,YI
CCNT INUE
K=K+KIC
CONT INUE
C-----CHANGES TABULAR DATA-----
241 IF(IPT.EQ.3) WRITE (6,401)
IF(IPT.EQ.2) WRITE (6,402)
WRITE(6,400)
CALL RDCCHAR ( IANS)
IF (( IANS.NE.IY).AND.( IANS.NE.IZ)) GO TO 245
GO TO 250
WRITE (6,500)
GO TO 241
CCNT INUE
245 IF ( IANS.EQ.IY) GO TC 290
IF ( IANS.EQ.IZ) GO TC 200
C-----CHANGES-----
290 CCNT INUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
WRITE(6,410)
WRITE(6,110)

```



```

292 CALL RCINT(IANS)
293 IF(IANS.GT.5.OR.IANS.LT.1) GO TO 292
294 GO TO 293
295 WRITE(6,510)
296 GO TO 291
297 CONTINUE
298 ICHG = 1
299 GO TO (251,252,253,254,200),IANS
300 -----
301 RETURN
302 -----
100 FORMAT(//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES TO:',//,10X,'1.
*LCOTTER / PAGE SIZE',//,10X,'2. GRAPH LIMITS',//,10X,
*3. HEADING',//,10X,'4. GAIN RANGE',//,10X,'5. NC ADDITIONAL CHANGES - PLOT ROOT-LOCUS',//,10X,
*6. NO CHANGES - EXIT ROOT-LCCUS PLOTTING ROUTINE',//)
110 FORMAT(//,5X,'ENTER OPTION NUMBER',//)
120 FORMAT(//,15X,'ROOT-LCCUS')
121 FORMAT(//,5X,'OPEN LCCP POLES AND ZEROS')
122 FORMAT(//,5X,'NOISE TF POLES AND ZEROS')
123 FORMAT(//,5X,'COMPENSATOR TF POLES AND ZEROS')
130 FORMAT(//,15X,'A DISPLA METAFILE HAS BEEN CREATED',//,
*15X,'CLEAR SCREEN TO CONTINUE',//,15X,'CONTINUE',//,15X)
300 FORMAT(//,10X,'INPUT NUMBER = ',I5)
301 FORMAT(//,10X,'OUTPUT NUMBER = ',I5)
302 FORMAT(//,10X,'NUMERATOR ORDER = ',I5)
303 FORMAT(//,10X,'DENOMINATOR ORDER = ',I5)
304 FORMAT(//,10X,'TRANSFER FUNCTION (DC) GAIN = ',D12.4,/)
305 FORMAT(//,10X,'GAIN = ',D12.4)
306 FORMAT(//,10X,'12.4,5X,D12.4,'IMAGINARY PART',//)
307 FORMAT(//,12X,'REAL LOCATIONS')
308 FORMAT(//,10X,'ZERO LOCATIONS')
309 FORMAT(//,15X,'NO REAL ZEROS',//)
310 FORMAT(//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES ? ',//,
*10X,'TYPE "YES" OR "NO"',//)
400 FORMAT(//,5X,'OUTPUT SENT TO PRINTER',//)
401 FORMAT(//,5X,'OUTPUT SENT TO LISTING FILE',//)
402 FORMAT(//,5X,'DO YOU DESIRE TO MAKE ANY CHANGES TO:',//,
*10X,'1. GAIN RANGE',//,
*10X,'2. NUMBER OF PLOTS CALCULATED',//,
*10X,'3. NC ADDITIONAL CHANGES - TABULATE DATA',//,
*10X,'4. NC EXIT ROOT-LCCUS WARNING',//,
*10X,'5. EXIT ROOT-LCCUS',//,
*10X,'ENTER "YES" OR "NO"',//)
500 FORMAT(//,5X,'OPEN LCCP TRANSFER FUNCTION',//)
510 FORMAT(//,10X,'IMPROPER DATA ENTRY',//)
710

```



```

711  FORMAT (5X,'NOISE TRANSFER FUNCTION',/)
712  FORMAT (5X,'COMPENSATOR TRANSFER FUNCTION',/)
C-----
      END
C=====
C SUBROUTINE EODE (NS,K2,K1,ITFX)
C INTERATIVELY PLOTS A BODE PLOT USING DISSPLA GRAPHICS GIVEN
C THE ZERO AND POLE LOCATIONS
C=====
      REAL*8 OLN,CLD,ORNUM,MA,OMEGA,QI,QR,QPSR,MAG,GAN,CO
      REAL*4 XDO,YDO,XNO,YNO,FMIN,FMAX,MMIN,MMAX,XAXIS,YAXIS,PAGEX,
      *PAGEY,XX,YY,PMIN,PMAX,MCYCLE,PCYCLE,GOL,FL,FU,PM,GM,
      *TH,TH1,TS,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
      COMPLEX*16 RTN,PLN,RTD,PLD,WN
      INTEGER NS,IPTR,K1,K2,NN,NL,LINES,IANS,ICHG,LINE,N,IPP,PTS,IPT
      *ORD,ITFX
      COMMON /GRAFF/ OLN(99,2,12,12),OLD(99,2),GAN(2,12),CC
      COMMON /SGRAF/ XDO(500),YDO(500),XNC(500),YNC(500)
      COMMON /SHEAD/ LINES(4,18),LINE(9)
      COMMON /FFREQ/ RTN(99),PLN(99),RTD(99),WN(500),QI(500),
      *QR(500),QPSR(500),MAG(500)
      COMMON /SCALE/ TS,TH,TH1,T1,T2,T3,T4,T5,T6,T7,T8,T9,T10
      DATA IV,Y,Y1,IZ,N/
C-----GRAPHIC OR TABULAR OUTPUT-----
200  CALL FRTCMS('CLRSCRN')
      WRITE(6,12C)
      IF(ITFX.EQ.1) WRITE(6,710)
      IF(ITFX.EQ.2) WRITE(6,711)
      IF(ITFX.EQ.3) WRITE(6,712)
      CALL PRTPLOT(IPP)
      GO TO (201,202,80),IPF
C-----PLOT SETUP-----
201  CONTINUE
      N=500
      ICHG=0
      IF(GAN(2,K2,K1).EQ.0.0) GO TO 95
      CALL FRTCMS('CLRSCRN')
      IF(ICHG.EQ.1) GO TO 41
      WRITE(6,120)
      IF(ITFX.EQ.1) WRITE(6,710)
      IF(ITFX.EQ.2) WRITE(6,711)
      IF(ITFX.EQ.3) WRITE(6,712)
      CALL GSETUP(IPTR,PAGEY)
      IF(ICHG.EQ.1) GO TO 50
      CALL FRTCMS('CLRSCRN')
      CALL FGRAF(FMIN,FMAX)
      IF(ICHG.EQ.1) GO TO 50

```

```

6C      CALL FRTCMS('CLRSCRN ')
      CALL GHEAD(NL)
      IF(ICHG.EQ.1) GO TO 90
      CONTINUE
-----SCALING CONSTANTS-----
      CALL PSCALE(PAGEX)
-----INPUT AFRAYS-----
      GOL = SNGL(GAN(2,K2,K1))
      FL = FMIN*Q.1
      FU = FMAX*Q.1
      XAXIS = PAGEX - 2.*T1
      YAXIS = PAGEY - T1 - T2
      FCYCLE = XAXIS/(ALOG10(FMAX)-ALOG10(FMIN))
      CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
      DO 5 I = 1, N
      CMGSA = CMAG(WN(I))
      XND(I) = SNGL(OMEGA)
      MA = 20.0*CMG10(MAG(I))
      YND(I) = SNGL(MA)
      YDO(I) = SNGL(QPSR(I))
      CONTINUE
      CALL MINMAX(N,YND,MMIN,MMAX)
      MMIN = AINT(MMIN) - 1.0
      MCYCLE = (MMAX-MMIN)/YAXIS
      MCYCLE = AINT(MCYCLE) + 1.0
      CALL MINMAX(N,YDO,PMIN,PMAX)
      PMIN = AINT(PMIN) - 1.0
      PCYCLE = (PMAX-PMIN)/YAXIS
      PCYCLE = AINT(PCYCLE) + 1.0
-----MAGNITUDE PLOT-----
      IF(IPTR.EQ.2) GO TO 10
      CALL TEK61E
      CALL PAGE(PAGEX,PAGEY)
      CALL HWR0T('AUTO')
      CALL HMSCAL('SCREEN')
      CALL NOBRDR
      GO TO 20
-----COMPRS-----
1C      CONTINUE
      CALL COMPRS
      CALL PAGE(PAGEX,PAGEY)
-----PLOT EXECUTION-----
2C      CONTINUE
      CALL NOCHECK
      CALL GRACE(C.)
      CALL PHYSOF(T1,T1)
      CALL AREA2D(XAXIS, YAXIS)

```

```

OPT09130
OPT09140
OPT09150
OPT09160
OPT09170
OPT09180
OPT09190
OPT09200
OPT09210
OPT09220
OPT09230
OPT09240
OPT09250
OPT09260
OPT09270
OPT09280
OPT09290
OPT09300
OPT09310
OPT09320
OPT09330
OPT09340
OPT09350
OPT09360
OPT09370
OPT09380
OPT09390
OPT09400
OPT09410
OPT09420
OPT09430
OPT09440
OPT09450
OPT09460
OPT09470
OPT09480
OPT09490
OPT09500
OPT09510
OPT09520
OPT09530
OPT09540
OPT09550
OPT09560
OPT09570
OPT09580
OPT09590
OPT09600

```

5

```

CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME('FREQUENCY - RADIANS PER SECOND $',100)
CALL HEIGHT(TH)
CALL YNAME('MAGNITUDE - DECIBELS $',100)
CALL XLOG(FMIN,FCYCLE,MMIN,MCYCLE)
CALL RLVEC(FMAX+FU,0.0,FMIN-FL,0.0,0.0000)
C-----HEADING-----
NN = NL + 1
IF(NL.EQ.0) GO TO 35
DO 35 I=1,NL
    CALL CLINE(I)
    CALL HEADIN(LINE,100,T3,NN).
35 CONTINUE
IF(ITFX.EQ.1)
    * CALL HEADIN('OPEN LOOP TF BODE MAGNITUDE$',100,T3,NN)
    * IF(ITFX.EQ.2)
    * CALL HEADIN('NOISE TF BODE MAGNITUDE$',100,T3,NN)
    * IF(ITFX.EQ.3)
    * CALL HEADIN('COMPENSATOR TF BODE MAGNITUDE$',100,T3,NN)
C-----MAGNITUDE-----
CALL CURVE(XNO,YNO,N,0)
C-----LEGEND-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAG('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----GRID-----
CALL DOT
CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(C)
C-----PHASE PLOT-----
IF(IPTR.EQ.2) GO TO 11
CALL TEK618
CALL PAGE(PAGEX,PAGEY)

```

```

CALL HWROT('AUTO')
CALL HWSCL('SCREEN')
CALL NCBDRP
GO TO 21
C-----COMFRS-----
11 CCNT INUE
CALL COMPRS
CALL PAGE(PAGEX,PAGEY)
21 CCNT INUE
C-----PLOT EXECUTION-----
CALL NOCHECK
CALL GFAC(C)
CALL PHYSOR(T1,T1)
CALL AREA2C(XAXIS, YAXIS)
CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME('FREQUENCY - RADIANS PER SECOND $',100)
CALL HEIGHT(TH)
CALL YNAME('PHASE - DEGREES $',100)
CALL XLOG(FMIN,FCYCLE,PMIN,PCYCLE)
CALL RLVEC(FMAX+FU,180.0,FMIN-FL,-180.0,0000)
CALL RLVEC(FMAX+FU,-180.0,FMIN-FL,-180.0,0000)
C-----HEADING-----
NN = NL + 1
IF(NL.EQ.0) GO TO 36
DO 36 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
36 CCNT INUE
IF(ITFX.EQ.1)
* CALL HEADIN('OPEN LOOP TF BODE PHASE$',100,T3,NN)
IF(ITFX.EQ.2)
* CALL HEADIN('NOISE TF BODE PHASE$',100,T3,NN)
IF(ITFX.EQ.3)
* CALL HEADIN('COMPENSATOR TF BODE PHASE$',100,T3,NN)
C-----PHASE-----
CALL CURVE(XNC,YDO,N,0)
C-----LEGEND-----
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL HEESAG('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL HEESAG('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')

```



```

CALL HEIGHT(TH1)
CALL MESSAGE('DC GAIN =',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GOL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----GRID-----
CALL DOT
CALL GRID(1,1)
CALL RESET('DOT')
CALL ENDPL(0)
IF(IPTR,NE,2) GO TO 75
CALL FRTCMS('CLRSCRN ')
WRITE(6,130)
C-----GAIN AND PHASE MARGIN-----
75 CALL FRTCMS('CLRSCRN ')
WRITE(6,135)
C-----GAIN MARGIN-----
DO 71 I = 2,N
IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72
IF(ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72
GO TO 71
GM = 0.0 - YNO(I)
WRITE(6,140) XNO(I), GM
CCONTINUE
72
71 C-----PHASE MARGIN-----
DO 76 I = 2,N
IF(YNO(I-1).GT.0.0.AND.YNO(I).LE.0.0) GO TO 77
GO TO 76
PM = 180.0 + YDO(I)
WRITE(6,150) XNO(I), PM
CCONTINUE
76 C-----CHANGES TO PLOT-----
90 CONTINUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,120)
IF(ITFX.EQ.1) WRITE(6,710)
IF(ITFX.EQ.2) WRITE(6,711)
IF(ITFX.EQ.3) WRITE(6,712)
WRITE(6,110)
WRITE(6,110)
CALL RLINT(IANS)
IF(IANS.GT.5.OR.IANS.LT.1) GO TO 15
GO TO 25
15 WRITE(6,510)
GO TO 30
25 CCONTINUE
ICHG = 1

```



```

C-----GO TO (40,50,60,70,200),IANS-----TABULAR DATA-----OPTI 1050
202 CONTINUEIF(GAN(2,K2,K1).EQ.0.0) GO TO 95OPTI 1060
    ICHG = 0OPTI 1070
    IF(ICHG.EQ.1) GO TO 249OPTI 1080
    IF(ICHG.EQ.1) GO TO 249OPTI 1090
    IF(ICHG.EQ.1) GO TO 249OPTI 1100
    IF(ICHG.EQ.1) GO TO 249OPTI 1110
    IF(ICHG.EQ.1) GO TO 249OPTI 1120
    IF(ICHG.EQ.1) GO TO 249OPTI 1130
    IF(ICHG.EQ.1) GO TO 249OPTI 1140
    IF(ICHG.EQ.1) GO TO 249OPTI 1150
    IF(ICHG.EQ.1) GO TO 249OPTI 1160
    IF(ICHG.EQ.1) GO TO 249OPTI 1170
    IF(ICHG.EQ.1) GO TO 249OPTI 1180
    IF(ICHG.EQ.1) GO TO 249OPTI 1190
    IF(ICHG.EQ.1) GO TO 249OPTI 1200
    IF(ICHG.EQ.1) GO TO 249OPTI 1210
    IF(ICHG.EQ.1) GO TO 249OPTI 1220
    IF(ICHG.EQ.1) GO TO 249OPTI 1230
    IF(ICHG.EQ.1) GO TO 249OPTI 1240
    IF(ICHG.EQ.1) GO TO 249OPTI 1250
    IF(ICHG.EQ.1) GO TO 249OPTI 1260
    IF(ICHG.EQ.1) GO TO 249OPTI 1270
    IF(ICHG.EQ.1) GO TO 249OPTI 1280
    IF(ICHG.EQ.1) GO TO 249OPTI 1290
    IF(ICHG.EQ.1) GO TO 249OPTI 1300
    IF(ICHG.EQ.1) GO TO 249OPTI 1310
    IF(ICHG.EQ.1) GO TO 249OPTI 1320
    IF(ICHG.EQ.1) GO TO 249OPTI 1330
    IF(ICHG.EQ.1) GO TO 249OPTI 1340
    IF(ICHG.EQ.1) GO TO 249OPTI 1350
    IF(ICHG.EQ.1) GO TO 249OPTI 1360
    IF(ICHG.EQ.1) GO TO 249OPTI 1370
    IF(ICHG.EQ.1) GO TO 249OPTI 1380
    IF(ICHG.EQ.1) GO TO 249OPTI 1390
    IF(ICHG.EQ.1) GO TO 249OPTI 1400
    IF(ICHG.EQ.1) GO TO 249OPTI 1410
    IF(ICHG.EQ.1) GO TO 249OPTI 1420
    IF(ICHG.EQ.1) GO TO 249OPTI 1430
    IF(ICHG.EQ.1) GO TO 249OPTI 1440
    IF(ICHG.EQ.1) GO TO 249OPTI 1450
    IF(ICHG.EQ.1) GO TO 249OPTI 1460
    IF(ICHG.EQ.1) GO TO 249OPTI 1470
    IF(ICHG.EQ.1) GO TO 249OPTI 1480
    IF(ICHG.EQ.1) GO TO 249OPTI 1490
    IF(ICHG.EQ.1) GO TO 249OPTI 1500
    IF(ICHG.EQ.1) GO TO 249OPTI 1510
    IF(ICHG.EQ.1) GO TO 249OPTI 1520
C-----TABULATE DATA-----CHANGES TABULAR DATA-----OPTI 1530
254 CCNT INUEZERO = 0.0OPTI 1540
    J1 = 0OPTI 1550
    GRD = DINT (GAN(1,K2,K1))OPTI 1560
    GAN = GAN(2,K2,K1)OPTI 1570
    CALL FRTCMS('CLRSCRN ')OPTI 1580
    WRITE(IPT,120)OPTI 1590
    IF(ITFX.EQ.0.1) WRITE(IPT,710)OPTI 1600
    IF(ITFX.EQ.0.2) WRITE(IPT,711)OPTI 1610
    IF(ITFX.EQ.0.3) WRITE(IPT,712)OPTI 1620
    WRITE(IPT,300) K1OPTI 1630
    WRITE(IPT,301) K2OPTI 1640
    WRITE(IPT,302) NSOPTI 1650
    WRITE(IPT,303) ORDOPTI 1660
    WRITE(IPT,304) GAN(2,K2,K1)OPTI 1670
    WRITE(IPT,305)OPTI 1680
    N = FREQ(NS,K2,K1,FMIN,FMAX,N)OPTI 1690
    CALL 220 I,NOPTI 1700
    CMAG = DIMAG(WN(I))OPTI 1710
    MA = 20.0 * DLOG10(MAG(I))OPTI 1720
    WRITE(IPT,330) OMEGA,MA,QPSR(I),QR(I),QI(I)OPTI 1730
    CONTINUEOPTI 1740
220 OPTI 1750
C-----IF(IPT.EQ.2) WRITE (6,401)OPTI 1760
    IF(IPT.EQ.2) WRITE (6,402)OPTI 1770
    WRITE(6,403)OPTI 1780
    CALL RECHAR (IANS)OPTI 1790

```

185


```

C-----GRAPHIC OR TABULAR OUTPUT-----OPTI2490
200 CALL FRTCMS('CLRSCRN ')OPTI2500
    WRITE(6,120)OPTI2510
    IF(ITFX.EQ.1) WRITE(6,710)OPTI2520
    IF(ITFX.EQ.2) WRITE(6,711)OPTI2530
    IF(ITFX.EQ.3) WRITE(6,712)OPTI2540
    CALL PRTPLOT(IPP)OPTI2550
    GO TO (201,202,80),IPPOPTI2560
C-----PLOT SETUP-----OPTI2570
201 CONTINUEOPTI2580
    N = 500OPTI2590
    ICHG = 0OPTI2600
    IF(GAN(2,K2,K1).EQ.0.0) GO TO 95OPTI2610
    CALL FRTCMS('CLRSCRN ')OPTI2620
    IF(ICHG.EQ.1) GO TO 41OPTI2630
    WRITE(6,120)OPTI2640
    IF(ITFX.EQ.1) WRITE(6,710)OPTI2650
    IF(ITFX.EQ.2) WRITE(6,711)OPTI2660
    IF(ITFX.EQ.3) WRITE(6,712)OPTI2670
    CALL GSETUP(IPTR,PAGEX,PAGEY)OPTI2680
    IF(ICHG.EQ.1) GO TO 90OPTI2690
    CALL FRTCMS('CLRSCRN ')OPTI2700
    CALL FGRAF(FMIN,FMAX)OPTI2710
    IF(ICHG.EQ.1) GO TO 90OPTI2720
    CALL FRTCMS('CLRSCRN ')OPTI2730
    CALL GHEAD(NL)OPTI2740
    IF(ICHG.EQ.1) GO TO 90OPTI2750
    CCNTINUEOPTI2760
C-----SCALING CONSTANTS-----OPTI2770
C-----INPUT ARRAYS-----OPTI2780
    CALL PSCALE(PAGEX)OPTI2790
    GOL = SNGL(GAN(2,K2,K1))OPTI2800
    CALL FREQ(NS,K2,K1,FMIN,FMAX,N)OPTI2810
    DO 5 I = 1,NOPTI2820
        XNO(I) = SNGL(QR(I))OPTI2830
        YNO(I) = SNGL(QI(I))OPTI2840
    CCNTINUEOPTI2850
    CALL MINMAX(N,XNO,QRMIN,QRMAX)OPTI2860
    CALL MINMAX(N,YNO,QIMIN,QIMAX)OPTI2870
C-----PLOT LIMITS-----OPTI2880
    XAXIS = PAGEX - 2.*T1OPTI2890
    YAXIS = PAGEY - T1 - T2OPTI2900
    IF(QRMIN.GT.-1.0) QRMIN = -1.0OPTI2910
    IF(QRMAX.LT.0.0) QRMAX = 0.0OPTI2920
    IF(QIMIN.LT.0.0) QIMIN = 0.0OPTI2930
    IF(QIMAX.LT.0.0) QIMAX = 0.0OPTI2940
    IF((QRMAX-QRMIN).GE.(QIMAX-QIMIN)) GO TO 6OPTI2950
    RL = QIMAX - QIMINOPTI2960

```


6	GO TO 7	QRMAL - QRMAL	OPTI 2970
7	B1 = .2	XAXIS	OPTI 2980
	B2 = .5	XAXIS	OPTI 2990
	B3 = .X	AXIS	OPTI 3000
	C3 = .5	YAXIS	OPTI 3010
	KK = 1	YAXIS	OPTI 3020
	IF(RL.GT.B1)	KK=2	OPTI 3030
	IF(RL.GT.B2)	KK=3	OPTI 3040
	IF(RL.GT.B3)	KK=4	OPTI 3050
	GO TO (11,12,13,14),KK		OPTI 3060
11	RL = B1		OPTI 3070
12	RL = B2		OPTI 3080
13	RL = B3		OPTI 3090
14	GO TO 16		OPTI 3100
17	IF(RL.GT.B1)		OPTI 3110
	IF(RL.GT.B2)		OPTI 3120
	IF(RL.GT.B3)		OPTI 3130
	IF(RL.GT.B1)		OPTI 3140
	IF(RL.GT.B2)		OPTI 3150
	IF(RL.GT.B3)		OPTI 3160
	IF(RL.GT.B1)		OPTI 3170
	IF(RL.GT.B2)		OPTI 3180
	IF(RL.GT.B3)		OPTI 3190
21	GO TO 17	(FLCAT(II))/(FLCAT(NI))	OPTI 3200
16	GO TO 18	RL/B3	OPTI 3210
18	IF((QRMAL-QRMAL)/RSTEP).GE.B2	GO TO 22	OPTI 3220
	XDIST = B2		OPTI 3230
	GO TO 23		OPTI 3240
22	XDIST = -QRMAL/RSTEP		OPTI 3250
23	IF((QRMAL-QRMAL)/RSTEP).GE.C2	GO TO 24	OPTI 3260
	YDIST = C2		OPTI 3270
	GO TO 26		OPTI 3280
24	YDIST = -QRMAL/RSTEP		OPTI 3290
26	IF(XDIST.GT.B3)	XDIST = B3	OPTI 3300
	IF(YDIST.GT.C3)	YDIST = C3	OPTI 3310
	THEFAC = 3.1415927/180.		OPTI 3320
C	IF(IPTREQ.2)	GO TO 10	OPTI 3330
	CALL TEK61		OPTI 3340
	CALL PAGE(PAGE,PAGE)		OPTI 3350
	CALL HWROT(AUTO)		OPTI 3360
	CALL HWSAL(SCREEN)		OPTI 3370
	CALL NCBDF		OPTI 3380
	GO TO 20		OPTI 3390
IC	CONTINUE		OPTI 3400
		COMPR	OPTI 3410
			OPTI 3420
			OPTI 3430
			OPTI 3440


```

20 C-----CALL CCMPRS
CALL PAGE(PAGEX,PAGEY)
CCNTINUE-----PLOT EXECUTION-----
CALL NOCHECK
CALL GRACE(C)
CALL PHYSOR(T1,T1)
CALL AREA2C(XAXIS, YAXIS)
CALL TRIPLX
CALL HEIGHT(TH)
CALL XNAME(' $',100)
CALL HEIGHT(TH)
CALL YNAME(' $',100)
C-----CALL POLAR(THFAC,RSTEP,XDIST,YDIST)-----POLAR PLOT-----
C-----NN = NL + 1
IF(NL.EQ.0) GO TO 35
DC 35 I=1,NL
CALL CLINE(I)
CALL HEADIN(LINE,100,T3,NN)
35 CONTINUE
IF(ITFX.EQ.0)
* IF(CALL HEADIN('OPEN LOOP TRANSFER NYQUIST$',100,T3,NN)
* IF(ITFX.EQ.2)
* IF(CALL HEADIN('NOISE TRANSFER NYQUIST$',100,T3,NN)
* IF(ITFX.EQ.3)
* IF(CALL HEADIN('COMPENSATOR TRANSFER NYQUIST$',100,T3,NN)
C-----XDO(1) = 0.0
XDO(2) = RL#2.0
YDO(1) = 0.0
YDO(2) = 0.0
DC 27 I=1,4
CALL CURVE(YEC,XDO,2,0)
YDO(1) = YEC(I) + 90.
CCNTINUE-----UNIT CIRCLE-----
27 C-----CALL DCT
CALL GRID(1,1)
XDO(1) = 0.0
YDO(1) = 1.0
DC 28 I=1,360
XDO(I) = XEC(I-1) + 1.0
YDO(I) = 1.0
CCNTINUE
CALL CURVE(XDO,YDO,360,0)
CALL RESET('DOT')
28

```

```

C-----CURVE PLOT-----OPTI 3930
DO 29 I = 1,N
XND(I) = SNGL(QPSR(I))
YNO(I) = SNGL(MAG(I))
25 CCNT INUE
CALL CURVE(XNO,YNO,N,O)
CALL FRAME
C-----LEGEND-----OPTI 3940
XX = XAXIS - T7
YY = YAXIS + T8
CALL HEIGHT(TH1)
CALL MESSAG('INPUT # = $',100,XX+T8,YY+T10)
CALL HEIGHT(TH1)
CALL INTNO(K1,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('OUTPUT # = $',100,XX+T8,YY+T9)
CALL HEIGHT(TH1)
CALL INTNO(K2,'ABUT','ABUT')
CALL HEIGHT(TH1)
CALL MESSAG('DC GAIN = $',100,XX+T8,YY+T8)
CALL HEIGHT(TH1)
CALL REALNC(GCL,-3,'ABUT','ABUT')
CALL BLREC(XX,YY,T5,T6,0.02)
C-----OPTI 4150
CALL ENDPL(C)
IF(IPTR.NE.2) GO TO 75
CALL FRTCMS('CLRSCRN ')
WRITE(6,13C)
C-----GAIN AND PHASE MARGIN-----OPTI 4160
DC 73 I = 1,N
CMGA = DMAG(WN(I))
XNO(I) = SNGL(OMEGA)
MA = 20.0*DLOG10(MAG(I))
YNO(I) = SNGL(MA)
YDO(I) = SNGL(QPSR(I))
72 CCNT INUE
CALL FRTCMS('CLRSCRN ')
WRITE(6,13E)
C-----GAIN MARGIN-----OPTI 4220
DC 71 I = 2,N
IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72
IF(ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72
GC TO 71
GM = 0.0 - YNO(I)
72 WRITE(6,14C) XNO(I), GM
CCNT INUE
C-----PHASE MARGIN-----OPTI 4330
DO 76 I = 2,N
OPTI 4340
OPTI 4350
OPTI 4360
OPTI 4370
OPTI 4380
OPTI 4390
OPTI 4400

```



```

WRITE(IPT,120) WRITE(IPT,710) OPTI 4890
IF(ITFX.EQ.2) WRITE(IPT,711) OPTI 4900
IF(ITFX.EQ.3) WRITE(IPT,712) OPTI 4910
WRITE(IPT,300) K1 OPTI 4920
WRITE(IPT,301) K2 OPTI 4930
WRITE(IPT,311) NS OPTI 4940
WRITE(IPT,310) CRD OPTI 4950
WRITE(IPT,320) GAN(2,K2,K1) OPTI 4960
WRITE(IPT,350) OPTI 4970
N = FREQ(NS,K2,K1,FMIN,FMAX,N) OPTI 4980
DO 220 I = 1,N OPTI 4990
  CMGMA = DIMAG(WN(I)) OPTI 5000
  WRITE(IPT,330) OMEGA,MAG(I),QPSR(I),QR(I),CI(I) OPTI 5010
  CCNT INUE OPTI 5020
220 -----CHANGES TABULAR DATA----- OPTI 5030
C----- OPTI 5040
IF(IPT.EQ.3) WRITE(6,401) OPTI 5050
IF(IPT.EQ.2) WRITE(6,402) OPTI 5060
CALL RCCHAR(IANS) OPTI 5070
IF((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245 OPTI 5080
GO TO 250 OPTI 5090
WRITE(6,500) OPTI 5100
245 -----CHANGES----- OPTI 5110
C----- OPTI 5120
CONT INUE OPTI 5130
IF(IANS.EC.IY) GO TC 290 OPTI 5140
IF(IANS.EC.IZ) GO TC 200 OPTI 5150
250 -----CHANGES----- OPTI 5160
C----- OPTI 5170
CONT INUE OPTI 5180
CALL FRTCMS('CLRSCRN ') OPTI 5190
WRITE(6,120) OPTI 5200
IF(ITFX.EQ.1) WRITE(6,710) OPTI 5210
IF(ITFX.EQ.2) WRITE(6,711) OPTI 5220
IF(ITFX.EQ.3) WRITE(6,712) OPTI 5230
WRITE(6,410) OPTI 5240
251 -----CHANGES----- OPTI 5250
CALL RCINT(IANS) OPTI 5260
IF(IANS.GT.5.OR.IANS.LT.1) GO TO 292 OPTI 5270
GO TO 293 OPTI 5280
WRITE(6,510) OPTI 5290
252 -----CHANGES----- OPTI 5300
CONT INUE OPTI 5310
ICHG = I OPTI 5320
GO TO (251,252,253,254,200),IANS OPTI 5330
C----- OPTI 5340
253 -----CHANGES----- OPTI 5350
C----- OPTI 5360
RETURN OPTI 5370
WRITE(6,520) OPTI 5380

```



```

C-----SCALING CONSTANTS-----OPTI 6330
CALL PSCALE(PAGEX)-----OPTI 6340
C-----INPUT ARRAYS-----OPTI 6350
      GOL = SNGL(GAN(2,K2,K1))
      CALL FREQ(NS,K2,K1,FMIN,FMAX,N)
      DO 5 I=1,N
      MA = 20.0*DLOG10(MAG(I))
      YDO(I) = SNGL(MA)
      XDO(I) = SNGL(QPSR(I))
      CONTINUE
      CALL MINMAX(N,YDO,MMIN,MMAX)
      CALL MINMAX(N,XDO,PMIN,PMAX)
C-----PLOT LIMITS-----OPTI 6410
      XAXIS = PAGEX -2.*T1
      YAXIS = PAGEY -T1 -T2
      IF(MMAX.LT.C.O) MMAX = 0.0
      IF(MMIN.GT.C.O) MMIN = 0.0
      IF(PMAX.LT.C.O) PMAX = 0.0
      IF(PMIN.GT.C.O) PMIN = 0.0
C-----TEK 618-----OPTI 6460
      IF(IPTR.EQ.2) GO TO 10
      CALL TEK618
      CALL PAGE(PAGEX,PAGEY)
      CALL HWRDT('AUTO')
      CALL HWSCAL('SCREEN')
      CALL NCBDR
      GO TO 20
C-----COMPRS-----OPTI 6470
1C  CCNT INUE-----OPTI 6480
      CALL CCMPRS
      CALL PAGE(PAGEX,PAGEY)
      CCNT INUE-----OPTI 6490
2C  CCNT INUE-----OPTI 6500
C-----PLOT EXECUTION-----OPTI 6510
      CALL NCHECK
      CALL GRACE(C)
      CALL PHYSOR(T1,T1)
      CALL AREA2D(XAXIS, YAXIS)
      CALL TRIPLEX
      CALL HEIGHT(TH)
      CALL XNAME('PHASE - DEGREES $',100)
      CALL HEIGHT(TH)
      CALL YNAME('MAGNITUDE - DECIBELS $',100)
      CALL GRAF(PMIN,SCALE,PMAX,MMIN,SCALE,MMAX)
      NA = NL + 1
      IF(NL.EQ.0) GO TO 35
      DO 35 I=1,NL
      CALL CLINE(I)
C-----HEADING-----OPTI 6520
      OPTI 6530
      OPTI 6540
      OPTI 6550
      OPTI 6560
      OPTI 6570
      OPTI 6580
      OPTI 6590
      OPTI 6600
      OPTI 6610
      OPTI 6620
      OPTI 6630
      OPTI 6640
      OPTI 6650
      OPTI 6660
      OPTI 6670
      OPTI 6680
      OPTI 6690
      OPTI 6700
      OPTI 6710
      OPTI 6720
      OPTI 6730
      OPTI 6740
      OPTI 6750
      OPTI 6760
      OPTI 6770
      OPTI 6780
      OPTI 6790
      OPTI 6800

```

```

35      CALL HEACIN(LINE,100,T3,NN)
      CONTINUE
      IF(ITFX.EQ.1)
      * CALL HEACIN('OPEN LOOP TRANSFER NICHOLS$',100,T3,NN)
      * IF(ITFX.EQ.2)
      * CALL HEACIN('NOISE TRANSFER NICHOLS$',100,T3,NN)
      * IF(ITFX.EQ.3)
      * CALL HEACIN('COMPENSATOR TRANSFER NICHOLS$',100,T3,NN)
C-----
      CALL CURVE(XCO,YDO,N,C)
C-----
      XX = XAXIS - T7
      YY = YAXIS + T8
      CALL HEIGHT(TH1)
      CALL MESSAGE('INPUT # = $',100,XX+T8,YY+T10)
      CALL HEIGHT(TH1)
      CALL INTNO(K1,'ABUT','ABUT')
      CALL HEIGHT(TH1)
      CALL MESSAGE('OUTPUT # = $',100,XX+T8,YY+T9)
      CALL HEIGHT(TH1)
      CALL INTNO(K2,'ABUT','ABUT')
      CALL HEIGHT(TH1)
      CALL MESSAGE('DC GAIN = $',100,XX+T8,YY+T8)
      CALL HEIGHT(TH1)
      CALL REALNC(GOL,-3,'ABUT','ABUT')
      CALL BLREC(XX,YY,T5,T6,0.02)
C-----
      CALL DOT
      CALL GRID(1,1)
      CALL RESET('DOT')
      CALL ENDPL(C)
      IF(IPTR.NE.2) GO TO 75
      CALL FRTCMS('CLRSCRN ')
      WRITE(6,130)
C-----
      DO 73 I = 1,N
      CMGA = DIMAG(WN(I))
      XNO(I) = SNGL(OMEGA)
      MA = 20.0*LOG10(MAG(I))
      YNO(I) = SNGL(MA)
      YDO(I) = SNGL(QPSR(I))
      CONTINUE
      CALL FRTCMS('CLRSCRN ')
      WRITE(6,135)
C-----
      DO 71 I = 2,N
      IF(ABS(YDO(I-1)).LT.180.0.AND.ABS(YDO(I)).GE.180.0) GO TO 72

```

72	IF (ABS(YDO(I-1)).GT.180.0.AND.ABS(YDO(I)).LE.180.0) GO TO 72	OPTI 7290
	GO TO 71	OPTI 7300
	GM = 0.0 - YNO(I)	OPTI 7310
71	WRITE (6,14C) XNO(I), GM	OPTI 7320
C	CONTINUE	OPTI 7330
	-----PHASE MARGIN-----	OPTI 7340
	DO 76 I = 2,N	OPTI 7350
	IF (YNO(I-1).GT.0.0.AND.YNO(I).LE.0.0) GO TO 77	OPTI 7360
77	GO TO 76	OPTI 7370
	PM = 180.0 + YDO(I)	OPTI 7380
76	WRITE (6,150) XNO(I), PM	OPTI 7390
C	CONTINUE	OPTI 7400
	WRITE (6,160)	OPTI 7410
	-----CHANGES TO PLOT-----	OPTI 7420
90	CONTINUE	OPTI 7430
	CALL FRTCMS('CLRSCRN')	OPTI 7440
	WRITE (6,12C)	OPTI 7450
	IF (ITFX.EQ.0.1) WRITE (6,710)	OPTI 7460
	IF (ITFX.EQ.0.2) WRITE (6,711)	OPTI 7470
	IF (ITFX.EQ.0.3) WRITE (6,712)	OPTI 7480
90	WRITE (6,100)	OPTI 7490
	WRITE (6,11C)	OPTI 7500
	CALL RDINT(IANS)	OPTI 7510
	IF (IANS.GT.5.0R.IANS.LT.1) GO TO 15	OPTI 7520
15	GO TO 25	OPTI 7530
	WRITE (6,51C)	OPTI 7540
25	GO TO 30	OPTI 7550
	CONTINUE	OPTI 7560
	ICFG = 1	OPTI 7570
C	GO TO (40,50,60,70,200),IANS	OPTI 7580
202	CONTINUE	OPTI 7590
	IF (IGAN(2,K2,K1).EQ.0.0) GO TO 95	OPTI 7600
251	ICHG = 0	OPTI 7610
	CALL FRTCMS('CLRSCRN')	OPTI 7620
	IF (ICHG.EQ.0.1) GO TO 249	OPTI 7630
	WRITE (6,120)	OPTI 7640
	IF (ITFX.EQ.0.1) WRITE (6,710)	OPTI 7650
	IF (ITFX.EQ.0.2) WRITE (6,711)	OPTI 7660
	IF (ITFX.EQ.0.3) WRITE (6,712)	OPTI 7670
249	CALL PRTR(IPT)	OPTI 7680
	IF (ICHG.EQ.0.1) GO TO 290	OPTI 7690
252	CALL FRTCMS('CLRSCRN')	OPTI 7700
	CALL FGRAF(FMIN,FMAX)	OPTI 7710
	IF (ICHG.EQ.0.1) GO TO 290	OPTI 7720
253	CALL FRTCMS('CLRSCRN')	OPTI 7730
	CALL NPTS(PTS)	OPTI 7740
	IF (ICHG.EQ.0.1) GO TO 290	OPTI 7750
		OPTI 7760


```

C-----TABULATE DATA-----OPTI 7770
254 CONTINUE                                OPTI 7780
    ZERO = 0.0                             OPTI 7790
    J1 = 0                                 OPTI 7800
    ORD = CINT (GAN(1,K2,K1))              OPTI 7810
    GAIN = GAN(2,K2,K1)                   OPTI 7820
    CALL FRTCMS('CLRSCRN ')               OPTI 7830
    WRITE(IPT,120) WRITE(IPT,710)          OPTI 7840
    IF(ITFX.EQ.1)                          OPTI 7850
    IF(ITFX.EQ.2) WRITE(IPT,711)           OPTI 7860
    IF(ITFX.EQ.3) WRITE(IPT,712)           OPTI 7870
    WRITE(IPT,300) K1                      OPTI 7880
    WRITE(IPT,301) K2                      OPTI 7890
    WRITE(IPT,311) NS                      OPTI 7900
    WRITE(IPT,310) ORD                     OPTI 7910
    WRITE(IPT,320) GAN(2,K2,K1)            OPTI 7920
    WRITE(IPT,350)                          OPTI 7930
    N = FPTS                               OPTI 7940
    CALL FREQ(NS,K2,K1,FMIN,FMAX,N)         OPTI 7950
    DO 220 I = 1,N                         OPTI 7960
    OMEGA = DIMAG(WN(I))                   OPTI 7970
    MA = 20.0 * DLOG10(MAG(I))             OPTI 7980
    WRITE(IPT,330) OMEGA,MA,QPSR(I),QR(I),QI(I)
    CONTINUE                                OPTI 7990
-----CHANGES TABULAR DATA-----OPTI 8000
220 IF(IPT.EQ.3) WRITE(6,401)              OPTI 8010
    IF(IPT.EQ.2) WRITE(6,402)             OPTI 8020
    WRITE(6,400)                           OPTI 8030
    CALL RDCCHAR (IANS)                    OPTI 8040
    IF ((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 245
    GO TO 250                               OPTI 8050
    WRITE(6,500)                            OPTI 8060
    GO TO 240                               OPTI 8070
    CONTINUE                                OPTI 8080
245 IF (IANS.EQ.IY) GO TC 290              OPTI 8090
    IF (IANS.EQ.IZ) GO TC 200              OPTI 8100
-----CHANGES-----OPTI 8110
245 CONTINUE                                OPTI 8120
    CONTINUE                                OPTI 8130
    CALL FRTCMS('CLRSCRN ')               OPTI 8140
    WRITE(6,120)                           OPTI 8150
    IF(ITFX.EQ.1) WRITE(6,710)             OPTI 8160
    IF(ITFX.EQ.2) WRITE(6,711)             OPTI 8170
    IF(ITFX.EQ.3) WRITE(6,712)             OPTI 8180
    WRITE(6,410)                            OPTI 8190
    WRITE(6,411)                            OPTI 8200
    CALL RDCINT(IANS)                       OPTI 8210
    IF(IANS.GT.5.0 OR IANS.LT.1) GO TO 292  OPTI 8220
    GO TO 293                               OPTI 8230
    CONTINUE                                OPTI 8240
291

```



```

510 FORMAT(/,5X,'***** ERROR *****',10X,'IMPPROPER DATA ENTRY',/) OPT1 8730
520 FORMAT(/,5X,'***** WARNING *****',10X,'NUMERATOR IS EQUAL TO ZERO') OPT1 8740
#C,/) OPT1 8750
710 FORMAT (5X,'OPEN LOCP TRANSFER FUNCTION',/) OPT1 8760
711 FORMAT (5X,'NOISE TRANSFER FUNCTION',/) OPT1 8770
712 FCRMAT (5X,'COMPENSATOR TRANSFER FUNCTION',/) OPT1 8780
C----- OPT1 8790
END OPT1 8800
C===== OPT1 8810
C===== OPT1 8820
C SUBROUTINE FREQ(NS,K2,K1,FMIN,FMAX,NN) OPT1 8830
C THE SUBROUTINE FREQ CALCULATES THE MAGNITUDE AND PHASE FOR THE OPT1 8840
C BODE, NYQUIST, AND NICHOLS RESPONSES OPT1 8850
C===== OPT1 8860
C REAL#8 OLD,CLN,MAG,QI,QR,QPSR,FRMAX,FRMIN,CYCLE,ZERO,XX,AA,GAIN, OPT1 8870
C *DEFLT,F,GAN,CC OPT1 8880
C REAL#4 FMAX,X,FMIN OPT1 8890
C COMPLEX*16 RTN,PLN,RTD,PLD,CPOLN,WN,CPOLC,CPOL OPT1 8900
C INTEGER I,J1,K2,K1,NS,IPHS,JJ,L,N,N1,N2,NN,ORDC OPT1 8910
C COMMON /GRAFF/ CLN(99,2),OLD(99,2),GAN(2,12,12),CC OPT1 8920
C COMMON /FFREQ/ RTN(99),PLN(99),PLD(99),WN(500),GI(500), OPT1 8930
C *GP(500),QPSF(500),MAG(500) OPT1 8940
C----- OPT1 8950
C----- OPT1 8960
C AA = DFLCAT(NN) ----- OPT1 8970
J1= 0 I = 1,NS OPT1 8980
DO 4 I = 1,NS OPT1 8990
RTN(I) = 0.C OPT1 9000
RTD(I) = 0.C OPT1 9010
CONTINUE OPT1 9020
DO 10 I = 1,NS OPT1 9030
IF((DABS(OLN(I,1,K2,K1)).LE.CO).AND.(DABS(CLN(I,2,K2,K1)).LE.CO)) OPT1 9040
* GO TO 20 OPT1 9050
J1= J1+1 OPT1 9060
RTN(J1) = DCMPLX(OLN(I,1,K2,K1),OLN(I,2,K2,K1)) OPT1 9070
CONTINUE OPT1 9080
RTD(I) = DCMPLX(OLD(I,1),OLD(I,2)) OPT1 9090
CONTINUE OPT1 9100
CRD = DINT(GAN(1,K2,K1)) OPT1 9110
IF(ORD.GT.J1) J1 = ORD OPT1 9120
C----- OPT1 9130
ZERO = 0.0 OPT1 9140
GAIN = GAN(2,K2,K1) OPT1 9150
IPHS = 0 OPT1 9160
C----- OPT1 9170
FRMAX = DBLE(FMAX) OPT1 9180
FRMIN = DBLE(FMIN) OPT1 9190
CYCLE = (DLOG10(FRMAX) - DLOG10(FRMIN))/(AA - 1.) OPT1 9200

```

```

DELTF = 10.*CYCLE
DO 80 I = 1,NN
IF (I.GT.1) GC TO 5
WN(I) = DCMPLX(ZERO,FRMIN)
GO TO 6
WN(I) = WN(I-1) * DELTF
-----POLYNOMIAL-----
CPOL = 1.
IF (J1.EQ.0) GO TO 7
DO 30 K = 1,J1
CPOL = CPOL*(WN(I)-RTN(K))/(WN(I)-RTD(K))
IF (J1.EQ.N) GO TO 8
JJ = JJ + 1
DO 40 K = JJ,NS
CPOL = CPOL/(WN(I)-RTD(K))
CPOL = CPOL*GAIN
-----MAGNITUDE-----
MAG(I) = CLABS(CPOL)
-----PHASE-----
QI(I) = DIMAG(CPOL)
QR(I) = CREAL(CPOL)
IF (QR(I)) 601,602,603
IF (QR(I)) 620,621,622
IF (QR(I)) 623,751,625
IF (QR(I)) 626,627,628
QPSR(I) = -180. + 57.29577951*DATAN(QI(I)/QR(I))
GO TO 555
CPSR(I) = -50.
GC TO 555
CPSR(I) = - 57.29577951*DATAN(-QI(I)/QR(I))
GO TO 555
CPSR(I) = 180.
GO TO 555
CPSR(I) = 0.0
GO TO 555
QPSR(I) = 180. - 57.29577951*DATAN(-QI(I)/QR(I))
GC TO 555
CPSR(I) = 50.
GO TO 555
QPSR(I) = 57.29577951*DATAN(QI(I)/QR(I))
GO TO 555
IF (IPHS.EQ.0) GO TO 640
IF ((QPSR(I-1).GT.90.)%AND.(QPSR(I).LT.-90.)) GO TO 630
IF ((QPSR(I-1).LT.-90.)%AND.(QPSR(I).GT.90.)) IPHS = 1
GC TO 640
N = I - 1
DO 650 L = 1,N
QPSR(L) = CPSR(L) - 360.

```

```

650 CCNTINUE
640 IF(IPHS.EQ.0) GO TO 700
    QPSR(I) = CFSR(I) - 360.
700 CCNTINUE
    EC
751 RETURN
    WRITE(6,752)
    RETURN
C-----
752 FORMAT (/, 5X, '***** PROGRAM FAILURE COMPUTING FREQUENCY *****')
C-----
    END
C=====
C=====
C=====
C SUBROUTINE GSETUP(IPTR,PAGEX,PAGEY)
C SUBROUTINE GSETUP INTERACTIVELY ALLOWS THE USER TO SELECT EITHER
C THE TEK618 CR VERSATEC PLOTTER AND SPECIFY A PAGE SIZE
C=====
    INTEGER IPTR, IANS
    REAL*8 ANSR
    REAL*4 PAGEX, PAGEY
C-----
C----- PLOTTER SELECTION-----
10 WRITE(6,100)
   WRITE(6,110)
   CALL RDINT(IANS)
   IF(IANS.LT.1.OR.IANS.GT.2) GO TO 20
   GO TO 30
20 WRITE(6,500)
   GO TO 10
30 CCNTINUE
   IPTR = IANS
   IF(IPTR.EQ.1) GO TO 80
C-----
C----- VRSTEC PAGE SIZE-----
   CALL FRTCMS('CLRSRN ')
C-----
C----- HEIGHT-----
40 WRITE(6,130)
   WRITE(6,140)
   CALL RDREAL(ANSR)
   IF(ANSR.LT.0.OR.IANS.GT.21.0) GO TO 45
   GO TO 50
45 WRITE(6,510)
   GO TO 40
50 CCNTINUE
   PAGEY = ANSR
C-----
C----- WIDTH-----
60 WRITE(6,150)
   CALL RDREAL(ANSR)
   IF(ANSR.LT.0.OR.IANS.GT.21.0) GO TO 65

```



```

65 GC TO 70
   WRITE(6,510)
70 GO TO 60
   CCNTINUE
   PAGEX = ANSF
   RETURN
C-----VRSTEC PAGE SIZE-----
80 CCNTINUE
   PAGEX = 11.0
   PAGEY = 8.5
   RETURN
C-----
100 FORMAT(/,5X,'PLOTTER SELECTION AND PAGE SIZE',/,5X,
*),(NOTE: PAGE IS 11.0 INCHES BY 8.5 INCHES WITH SELECTION OF TEK618
*),/,/)
110 FORMAT(/,10X,'WHICH PLCTTER DO YOU DESIRE:',/,15X,'1. TEK618',/,
*15X,'2. VERSATEC',/,10X,'CHOCSE OPTION 1 CR 2')
130 FCORMAT(/,10X,' PAGE SIZE (MAXIMUM = 21.0 INCHES BY 21.0 INCHES)',
* ,/)
140 FORMAT(/,15X,'HEIGHT =',/,/)
150 FORMAT(/,15X,'WIDTH =',/,/)
160 FORMAT(/,15X,'***** ERROR *****',/,/)
170 FORMAT(/,15X,'***** ERROR *****',/,/)
   END
C=====
C-----
C-----SUBROUTINE GGRAF(XMIN,XMAX,YMIN,YMAX)
C-----SUBROUTINE GGRAF INTERACTIVELY ALLOWS THE USER TO SELECT PLOTTING
C-----LIMITS FOR USE WITH DISSPLA GRAPHICS
C=====
   REAL XMIN,XMAX,YMIN,YMAX,ANSR
   DATA IV,'Y',/,12,'N',/
C-----
40 CCNTINUE
   WRITE(6,100)
   WRITE(6,105)
C-----
C-----WRITE(6,110)
   CALL RDCREAL(ANSR)
   XMIN = ANSR
C-----
C-----WRITE(6,120)
   CALL RDCREAL(ANSR)
   XMAX = ANSR
C-----
C-----WRITE(6,125)
   YMIN = ANSR
C-----
C-----WRITE(6,130)

```



```

DATA IY/Y/Y/,IZ/N/
-----
4C CCNT INUE
   WRITE(6,10C)
   WRITE(6,10E)
-----
5C WRITE(6,11C)
   CALL RCREAL(ANSR)
   FMIN = ANSR
   IF(FMIN.GT.C.0) GO TC 60
   WRITE(6,110)
   GO TO 5C
-----
6C WRITE(6,120)
   CALL RCREAL(ANSR)
   FMAX = ANSR
   IF(FMAX.GT.C.0) GO TC 70
   WRITE(6,120)
   GO TO 6C
-----
7C CALL FRTCMS('CLRSCRN ')
   WRITE(6,10C)
   WRITE(6,10E)
   WRITE(6,150) FMIN
   WRITE(6,160) FMAX
   WRITE(6,19C)
   CALL RDCCHAR(IANS)
   IF((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 1C
   GO TO 20
   WRITE(5,50C)
   GO TO 30
   CCNT INUE
   IF(IANS.EQ.IY) GO TC 40
   RETURN
-----
100 FORMAT(/,5X,'PLOTTING LIMITS FOR GRAPH',/)
110 FORMAT(10X,'FREQUENCY RANGE - RADIANS PER SECOND',/)
1150 FORMAT(15X,'MINIMUM FREQUENCY = ',/,/)
1160 FORMAT(15X,'MAXIMUM FREQUENCY = ',/,15.3,/)
1190 FORMAT(15X,'MINIMUM FREQUENCY = ',/,15.3,/)
1190 FORMAT(/,5X,'DO YOU DESIRE TO MAKE ANY CHANGES ?',/,10X,
   * TYPEAT('YES' OR "NO",/)
500 FORMAT(/,5X,'WARNING',/,5X,'IMPROPER DATA ENTRY',/,
   * ENTER "YES" OR "NO",/)
510 FORMAT(/,5X,'WARNING',/,5X,'IMPROPER DATA ENTRY',/,
   * FREQUENCY MUST BE GREATER THAN ZERO.',/)
   END
-----
OPT211130
OPT211140
OPT211150
OPT211160
OPT211170
OPT211180
OPT211190
OPT211200
OPT211210
OPT211220
OPT211230
OPT211240
OPT211250
OPT211260
OPT211270
OPT211280
OPT211290
OPT211300
OPT211310
OPT211320
OPT211330
OPT211340
OPT211350
OPT211360
OPT211370
OPT211380
OPT211390
OPT211400
OPT211410
OPT211420
OPT211430
OPT211440
OPT211450
OPT211460
OPT211470
OPT211480
OPT211490
OPT211500
OPT211510
OPT211520
OPT211530
OPT211540
OPT211550
OPT211560
OPT211570
OPT211580
OPT211590
OPT211600

```

```

C=====OPT21610
C=====OPT21620
C=====OPT21630
C      SUBROUTINE RANGK(KMIN,KMAX)
C      SUBROUTINE RANGK INTERACTIVELY ALLOWS THE USER TO SELECT GAIN
C      LIMITS FOR USE WITH ROOT-LOCUS PLOTS
C=====OPT21640
C=====OPT21650
C=====OPT21660
C      REAL*8 KMIN,KMAX
C      DATA IY,'Y',IZ,'N'/
C=====OPT21670
C=====OPT21680
C      CCNTINUE
C      WRITE(6,100)
C=====OPT21690
C=====OPT21700
C=====OPT21710
C=====OPT21720
C      WRITE(6,110)
C      CALL RCREAL(ANSR)
C      KMIN = ANSR
C=====OPT21730
C=====OPT21740
C=====OPT21750
C      WRITE(6,120)
C      CALL RCREAL(ANSR)
C      KMAX = ANSR
C=====OPT21760
C=====OPT21770
C=====OPT21780
C=====OPT21790
C      CALL FRTCMS('CLRSCRN ')
C      WRITE(6,100)
C      WRITE(6,150) KMIN
C      WRITE(6,160) KMAX
C      WRITE(6,190)
C      CALL RDCCHAR(IANS)
C      IF((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 10
C      GO TO 20
C      WRITE(5,500)
C      GO TO 30
C      CCNTINUE
C      IF(IANS.EQ.IY) GO TO 40
C      RETURN
C=====OPT21800
C=====OPT21810
C=====OPT21820
C=====OPT21830
C=====OPT21840
C=====OPT21850
C=====OPT21860
C=====OPT21870
C=====OPT21880
C=====OPT21890
C=====OPT21900
C=====OPT21910
C=====OPT21920
C=====OPT21930
C=====OPT21940
C      FORMAT('/',5X,'GAIN RANGE FOR ROOT-LOCUS PLOT',//)
C      FORMAT(15X,'MINIMUM GAIN = ?',//)
C      FORMAT(15X,'MAXIMUM GAIN = ?',//)
C      FORMAT(15X,'MINIMUM GAIN = ',E15.3,//)
C      FORMAT(15X,'MAXIMUM GAIN = ',E15.3,//)
C      FORMAT('/',5X,'DO YOU DESIRE TO MAKE ANY CHANGES ?',/,10X,
C      *,'TYPE "YES" OR "NO".',//)
C      *,'WARNING 警告',//)
C      FORMAT('/',5X,'ENTER "YES" OR "NO".',//)
C      *3X,'END
C=====OPT21950
C=====OPT21960
C=====OPT21970
C=====OPT21980
C=====OPT21990
C=====OPT22000
C=====OPT22010
C=====OPT22020
C=====OPT22030
C=====OPT22040
C=====OPT22050
C=====OPT22060
C      SUBROUTINE GHEAD(NL)
C      SUBROUTINE GHEAD INTERACTIVELY ALLOWS THE USER TO INPUT THREE
C=====OPT22070
C=====OPT22080

```

```

C=====
C LINES OF HEADING FOR USE WITH DISSPLA GRAPHICS
C=====
C INTEGER NL, LINES, I, IANS, LINE
C COMMON /SHAD/ LINES(4,18), LINE(9)
C DATA IY/'Y', IZ/'N'/
C-----
1C WRITE(6,10C)
   WRITE(6,11C)
   CALL RCINT(IANS)
   IF(IANS.GT.3) GO TO 30
   GO TO 20
2C WRITE(6,51C)
   GO TO 10
3C CONTINUE
   IF(IANS.LT.0) GO TO 35
   GO TO 25
35 WRITE(6,520)
   GO TO 10
4C CONTINUE
   NL = IANS
   IF(NL.EQ.0) GO TO 70
   WRITE(6,120)
   DO 50 I=1,NL
     WRITE(6,130) I
     CALL RLIN(I)
   CONTINUE
50 -----CORRECTIONS-----
8C CALL FRTCMS('CLRSCRN ')
   WRITE(6,100)
   DO 60 I=1,NL
     WRITE(6,140) I, (LINES(I,J), J=1,9)
   CONTINUE
60 WRITE(6,150)
63 CALL RDCHAR(IANS)
   IF((IANS.NE.IY).AND.(IANS.NE.IZ)) GO TO 61
   GO TO 62
61 WRITE(6,500)
   GO TO 63
62 CONTINUE
   EQ.IZ) GO TO 70
77 IF(IANS.160)
   WRITE(6,160)
   CALL RCINT(IANS)
   CALL IF((IANS.LT.1).OR.(IANS.GT.NL)) GO TO 75
   GO TO 76
75 CONTINUE
   EQ.530)
   WRITE(6,77)
   GO TO 77
76 CONTINUE

```



```

C-----STOP
5C  FORMAT(1X,'WARNING: NULL STRINGS ARE NOT ALLCWD, ENTER',
    *1X,'CHARACTER VALUES.')
```

OPT233050
OPT233060
OPT233070
OPT233080
OPT233090
OPT233100
OPT233110
OPT233120
OPT233130
OPT233140
OPT233150
OPT233160
OPT233170
OPT233180
OPT233190
OPT233200
OPT233210
OPT233220
OPT233230
OPT233240
OPT233250
OPT233260
OPT233270
OPT233280
OPT233290
OPT233300
OPT233310
OPT233320
OPT233330
OPT233340
OPT233350
OPT233360
OPT233370
OPT233380
OPT233390
OPT233400
OPT233410
OPT233420
OPT233430
OPT233440
OPT233450
OPT233460
OPT233470
OPT233480
OPT233490
OPT233500
OPT233510
OPT233520

```

6C  FORMAT(///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
7C  FORMAT(9A4)
    END
C=====
C  SUBROUTINE CLINE(ILN)
C  SUBROUTINE PLACES A DCLLAR SIGN AT THE END OF A CHARACTER STING
C  ENTERED FROM THE KEYBOARD AND PLACES THE RESULTING STRING INTO
C  A 1-D ARRAY TC BE USED BY DISSPLA
C=====
C  INTEGER I,ILN,LINES,J,LINE,DOLLAR,BLANK
C  COMMON /SHEAD/ LINES(4,18),LINE(9)
C-----
C  DATA DOLLAR/'$ ',BLANK/' '
C  DO 10 I=1,5
    J=1
    LINE(I)=LINES(ILN,I)
    IF(LINE(I).EQ.BLANK) GO TO 20
1C  CCNT=INUE DCLLAR
2C  LINE(J)=DCLLAR
    RETURN
    END
C=====
C  SUBROUTINE RDREAL -- INTERACTIVELY READS A REAL NUMBER REPLY
C  INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL
C  STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.
C=====
C  SUBROUTINE RDREAL (ANSR)
C  REAL*8 ANSR
C  INTEGER COUNT
C-----
1C  COUNT=0
    CCNT=INUE
    CCNT=COUNT+1
    IF (COUNT.LT.3) GO TO 20
    WRITE (6,6C)
    GO TO 40
2C  CCNT=INUE
    READ (5,*,END=30,ERR=30) ANSR
    RETURN
3C  REWRITE (6,5C)
    GO TO 10
```

```

4C      CONTINUE                                OPT233530
      STOP                                     OPT233540
-----
5C      FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.) OPT233550
6C      FORMAT (///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/) OPT233560
      END                                     OPT233570
-----
C===== OPT233580
C===== OPT233590
C===== OPT233600
C===== OPT233610
C      SUBROUTINE RDINT -- INTERACTIVELY READS AN INTEGER REPLY          =OPT233620
C      INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY ENTERS A NULL  =OPT233630
C      STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY.           =OPT233640
C===== OPT233650
C      SUBROUTINE RDINT (IANS)
C      INTEGER COUNT,IANS
C=====
C      COUNT=0                                OPT233660
C      CONTINUE                                OPT233670
C      COUNT=COUNT+1                          OPT233680
C      IF (COUNT.LT.3) GO TO 20              OPT233690
C      WRITE (6,60)                           OPT233700
C      GO TO 40                                OPT233710
2C      CONTINUE                                OPT233720
C      READ (5,*,END=30,ERR=30) IANS          OPT233730
C      RETURN 5                                OPT233740
3C      REWIND 5                                OPT233750
C      WRITE (6,50)                            OPT233760
C      GO TO 10                                OPT233770
4C      CONTINUE                                OPT233780
C      STOP                                     OPT233790
-----
C===== OPT233800
C===== OPT233810
C===== OPT233820
C===== OPT233830
5C      FORMAT (1X,64HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER A NUMERICAL VALUE.) OPT233840
6C      FORMAT (///,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/) OPT233850
      END                                     OPT233860
-----
C===== OPT233870
C===== OPT233880
C===== OPT233890
C      SUBROUTINE RDCHAR -- INTERACTIVELY READS A CHARACTER STRING REPLY =OPT233900
C      (.YES. OR .NO.) INTO A FORTRAN PROGRAM. IF THE USER INADVERTENTLY =OPT233910
C      ENTERS A NULL STRING THE S/R ISSUES A WARNING AND ALLOWS A RECOVERY=OPT233920
C===== OPT233930
C      SUBROUTINE RDCHAR (IANS)
C      INTEGER COUNT,IANS
C=====
C      COUNT=0                                OPT233940
C      CONTINUE                                OPT233950
C      COUNT=COUNT+1                          OPT233960
C      IF (COUNT.LT.3) GO TO 20              OPT233970
1C      CONTINUE                                OPT233980
C      CONTINUE                                OPT233990
C      CONTINUE                                OPT234000

```

```

      WRITE (6,6C)
      GO TO 40
      CCNTINUE
      REWIND 5
      READ (5,70,END=30,ERR=30) IANS
      RETURN 5
      REWIND 5
      WRITE (6,5C)
      GO TO 10
      CCNTINUE
      STOP
C-----
5C  FORMAT (1X,60HWARNING:  NULL STRINGS ARE NOT ALLOWED, ENTER "YES"
1CR "NO".)
6C  FORMAT (//,2X,42HPROGRAM KILLED - TWO NULL STRINGS ENTERED ,/)
7C  FORMAT (A1)
      END
C=====
C  SUBROUTINE MINMAX(N,AA,MIN,MAX)
C  SUBROUTINE MINMAX FINDS THE MINIMUM AND MAXIMUM VALUES FOR A ONE
C  DIMENSIONAL ARRAY
C=====
      REAL*4 MIN,MAX,AA
      INTEGER N
      DIMENSION AA(N)
C-----
      MAX = AA(1)
      MIN = AA(1)
      DO 10 I = 1,N
      IF(AA(I).GT.MAX) GC TO 5
      IF(AA(I).LT.MIN) GC TO 5
      GO TO 10
      MAX = AA(I)
      GO TO 10
      MIN = AA(I)
      CONTINUE
      RETURN
      END
C=====
C  SUBROUTINE MAKPOL(N,R,C)
C  SUBROUTINE MAKPOL COMPUTES THE COMPLEX COEFFICIENTS OF AN N-TH
C  DEGREE POLYNOMIAL GIVEN N COMPLEX ROOTS OF THE POLYNOMIAL
C=====
      COMPLEX*16 P(N),C(N)
      IF(N.LE.0) RETURN
      DO 10 I=1,N

```



```

C=====OPT24970
C=====OPT24980
C=====OPT24990
C      SUBROUTINE PRTR(IPT)
C      SUBROUTINE PRTR INTERACTIVELY ALLOWS THE USER TO SELECT THE
C      THE OUTPUT DEVICE FOR TABULAR DATA.
C=====OPT25000
C=====OPT25010
C=====OPT25020
C=====OPT25030
C      INTEGER IPT, IANS
C-----OPT25040
C      WRITE(6,100)
C      CALL RDINT(IANS)
C      IF(IANS.GT.3.OR.IANS.LT.1) GO TO 15
C      GO TO 25
C      WRITE(6,510)
C      GO TO 30
C      CONTINUE
C      GO TO (40,50,60), IANS
C      IPT = 6
C      GO TO 80
C      IPT = 3
C      GO TO 80
C      IPT = 2
C      GO TO 80
C-----OPT25100
C      RETURN
C-----OPT25200
C-----OPT25210
C-----OPT25220
C      FORMAT(/,5X,'DO YOU DESIRE TABULAR OUTPUT TO GO TO:',/,
C      *10X,'1.',SCREEN',/,
C      *10X,'2.',PRINTER',/,
C      *10X,'3.',DISK(OPGRAPH LISTING)',/,
C      FORMAT(/,5X,'ENTER OPTION NUMBER',/,
C      FORMAT(/,5X,'***** ERROR *****',10X,'IMPROPER DATA ENTRY',/)
C-----OPT25240
C-----OPT25250
C-----OPT25260
C-----OPT25270
C-----OPT25280
C-----OPT25290
C-----OPT25300
C-----OPT25310
C-----OPT25320
C-----OPT25330
C      SUBROUTINE APTS(PTS)
C      SUBROUTINE APTS INTERACTIVELY ALLOWS THE USER TO SELECT THE NUMBER
C      OF POINTS TO BE CALCULATED FOR TABULAR DATA
C=====OPT25340
C=====OPT25350
C=====OPT25360
C=====OPT25370
C      INTEGER PTS, IANS
C-----OPT25380
C-----OPT25390
C      WRITE(6,100)
C      CALL RDINT(IANS)
C      IF(IANS.GT.500.OR.IANS.LT.1) GO TO 15
C      GO TO 25

```

```

15  WRITE(6,510)
25  GC TO 30
    CCNT INUE
    PTS = IANS
C-----
    RETURN
C-----
100  FORMAT(/,5X,'HOW MANY POINTS DO YOU WANT TO TABULATE?',/,
    *10X,'(500 IS THE MAXIMUM)',/,)
110  FORMAT(/,5X,'ENTER NUMBER OF POINTS',/,)
510  FORMAT(/,5X,'***** ERROR ***** MUST ENTER BETWEEN 1 AND 50
    *0 POINTS ',/,)
C-----
    END
C=====
C=====
C-----
    SUBROUTINE PSCALE(PAGEX)
    SUBROUTINE PSCALE AUTOMATICALLY SCALES DISSPLA FLOTS
C=====
    REAL*4 PAGEX,TS,TH,IH,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
    COMMON /SCALE/ TS,TH,IH,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
C-----
    TS = PAGEX/11.0
    TH = 0.14*TS
    IH = 0.10*TS
    I1 = 0.60*TS
    I2 = 1.60*TS
    I3 = 1.20
    I4 = 0.30*TS
    I5 = 2.10*TS
    I6 = 0.19
    I7 = 2.00*TS
    I8 = 0.20*TS
    I9 = 2.0*TS
    I10 = 3.0*TS
    IF(PAGEX.LT.6.0) T3 =1.0
    RETURN
C-----
    END

```

```

OPT25450
OPT25460
OPT25470
OPT25480
OPT25490
OPT25500
OPT25510
OPT25520
OPT25530
OPT25540
OPT25550
OPT25560
OPT25570
OPT25580
OPT25590
OPT25600
OPT25610
OPT25620
OPT25630
OPT25640
OPT25650
OPT25660
OPT25670
OPT25680
OPT25690
OPT25700
OPT25710
OPT25720
OPT25730
OPT25740
OPT25750
OPT25760
OPT25770
OPT25780
OPT25790
OPT25800
OPT25810
OPT25820
OPT25830

```

APPENDIX D
OPGRAPH LISTING

This portion of the thesis contains a sample of tabular output sent to a disk (OPGRAPH LISTING)

BCCE PLOT COMPENSATOR TRANSFER FUNCTION

```

INPUT NUMBER = .1
OUTPUT NUMBER = 1
DENOMINATOR ORDER = 2
NUMERATOR CRDER = 1
TRANSFER FUNCTION (DC) GAIN = -0.1135D+06

FREQUENCY (RAD/SEC) 0.1000D+02
0.1778D+02
0.3162D+02
0.5623D+02
C.1000D+02

MAGNITUDE (DECIBELS) 0.4815D+02
0.5106D+02
0.5554D+02
0.5891D+02
0.5956D+02

PHASE (DEGREES) -0.1482D+03
-0.1401D+03
-0.1423D+03
-0.1629D+03
-0.2025D+03

REAL PART -0.2171D+03
-0.2740D+03
-0.4466D+03
-0.8431D+03
-0.8786D+03

IMAGINARY PART -0.1346D+03
-0.2292D+03
-0.3456D+03
-0.2598D+03
0.3637D+03

```

LIST OF REFERENCES

1. Hall, W. E., Computational Methods for the Synthesis of Rotary-Wing VTOL Aircraft Control Systems, Ph.D. Dissertation, Stanford Univ., Aug. 1971.
2. Walker, R. A., User's Manual for OPTSYS 4 at SCIP, Stanford Univ., Aero/Astro Dept., Dec. 1979.
3. Liu, G., User's Manual for OPTSYS 5 at CIT, Stanford Univ., Aero/Astro Dept., Aug. 1982.
4. Hoden, J. G., Interactive Implementation of the Optimal Systems Control Design Program (OPTSYSX) on the IBM/3033, MS Thesis, Naval Postgraduate School, Monterey CA.
5. Diel, H. A., Development of Graphical Time Response using the OPTSYSX Program, MS Thesis, Naval Postgraduate School, Monterey CA.
6. Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Hemisphere Pub. Co., 1969, (2nd Printing, 1975).
7. Research and Educational Association, Problem Solver in Automatic Control Systems/Robotics, 1982.
8. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, 1972.

BIBLIOGRAPHY

D'Azzo, J.J. and Houpis, C.H., Linear Control System Analysis and Design: Conventional and Modern, McGraw-Hill, 1981.

DiStefano III, J.J., Stupperud, A.R., and Williams, I.J., Feedback and Control Systems, Schaum's Outline Series, McGraw-Hill, 1967.

Lipschutz, S. and Poe, A., Programming with FORTRAN, Schaum's Outline Series, McGraw-Hill, 1978.

Melsa, J.L. and Jones, S.K., Computer Programs for Computational Assistance in the Study of Linear Control Theory, McGraw-Hill, 1973.

INITIAL DISTRIBUTION LIST

	No.	Copies
1. Defense Technical Information Center Cameron Station Alexandria Va 22314	2	
2. Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, Ca 93943	1	
3. Library, Code 0142 Naval Postgraduate School Monterey Ca 93943	2	
4. Professor D.J. Collins Code 67Co Department of Aeronautics Naval Postgraduate School Monterey, Ca 93943	5	
5. CDR V.C. Gordon 315 Alemeda Blvd. Coronado, Ca 92118	1	
6. Professor A.E. Bryson Department of Aeronautics and Astronautics Stanford University Stanford, Ca 94305	1	
7. LT M.H. Laptas 39 Valley Heights Holyoke, Ma 01040	5	



210349

Thesis

L2673 Laptas

c.1 Development of graphical Pole-Zero, Root-Locus, Bode, Nyquist, and Nichols responses using the OPTSYSX Program.

210349

Thesis

L2673 Laptas

c.1 Development of graphical Pole-Zero, Root-Locus, Bode, Nyquist, and Nichols responses using the OPTSYSX Program.

thesL2673

Development of graphical Pole-zero, Root



3 2768 002 12247 5

DUDLEY KNOX LIBRARY